IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the Application of:

Bardzil et al.

Serial No.: 10/601,158

Filed: 06-19-2003

Atty. File No.: 4366-133

For: DETECTION OF LOAD
BALANCED LINKS IN
INTERNET PROTOCOL
NETWORKS

Group Art Unit: 2157

Examiner: NANO, SARGON N

Confirmation No.: 6927

REQUEST FOR CONSIDERATION OF REFERENCES SUBMITTED IN INFORMATION DISCLOSURE STATEMENT OF JUNE 19, 2003

Electronically Submitted

Mail Stop Issue Fee
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

On June 19, 2003, Applicants submitted an Information Disclosure Statement (IDS), in the above-identified patent application. The Information Disclosure Statement, is attached hereto as Exhibit A.

On June 14, 2007, the Examiner issued an Official Action for the above-identified case. The Official Action was accompanied by a List Of References Cited By Applicant And Considered By Examiner. This list included the aforementioned Information Disclosure Statement of June 19, 2003. The considered version of the IDS does not indicate that References AV and AW (i.e., US-2003/0046427 and US-10/127,938) have been considered, as these references are not initialed or marked in any other fashion. The pertinent page from the considered List Of References Cited By Applicant And Considered By Examiner, is attached hereto as Exhibit B.

On January 17, 2008, the Examiner issued a Notice of Allowance for the above-identified case. The Notice of Allowance was accompanied by a "correction" List Of References Cited By Applicant And Considered By Examiner. This list included the aforementioned Information Disclosure Statement of June 19, 2003. The corrected considered version of the IDS also does not indicate that References AV and AW have been considered, as these references are not initialed or marked in any other fashion. The pertinent page from the considered List Of References Cited By Applicant And Considered By Examiner, is attached hereto as Exhibit C.

Due to the fact that the Information Disclosure Statement of June 19, 2003, was correctly submitted to the USPTO, Applicants hereby respectfully request that References AV and AW listed in that IDS, be considered by the Examiner.

For the Examiner's convenience, References AV and AW are attached hereto as Exhibit D.

Although no fees are believed due in connection with this communication, please charge any fees deemed necessary to Deposit Account No. 19-1970. If additional information is required please contact the undersigned.

Respectfully submitted,

SHERIDAN ROSS P.C.

By: Douglas Swartz
Douglas W. Swartz
Registration No. 37739
1560 Broadway, Suite 1200
Denver, Colorado 80202-5141
(303) 863-9700

Date: April 15, 2008

EXHIBIT A

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the Application of:

BARDZIL et al.

Serial No.: Not Yet Assigned

Filed: Herewith

Atty. File No.: 4366-133

For: "DETECTION OF LOAD BALANCED LINKS IN INTERNET PROTOCOL NETWORKS"

Group Art Unit:

Examiner:

INFORMATION DISCLOSURE STATEMENT

"EXPRESS MAIL" MAILING LABEL NUMBER: EV331286859US
DATE OF DEPOSIT: 6/19/03

I HEREBY CERTIFY THAT THIS CORRESPONDENCE IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE "EXPRESS MAIL POST OFFICE TO ADDRESSEE" SERVICE UNDER 37 C.F.R. 1.10 ON THE DATE INDICATED ABOVE AND IS ADDRESSED TO THE COMMISSIONER FOR PATENTS, P.O. BOX 1450, ALEXANDRIA, VA 22313-1450.

TYPED OR PRINTED NAME: Amy S. Duarte

SIGNATURE: Amy Duarte

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The references cited on attached Form PTO-1449 are being called to the attention of the Examiner. Copies of the cited references:

☒ Are enclosed herewith.

□ Are not enclosed, in accordance with 37 C.F.R. 1.98(d), because the references were submitted to the U.S. Patent and Trademark Office in prior application Serial No. ________________ filed ____________, which is relied upon for an earlier filing date under 35 U.S.C. § 120.

□ To the best of applicants' belief, the pertinence of the foreign-language references are believed to be summarized in the attached English abstracts and in the figures, although applicants do not necessarily vouch for the accuracy of the translation.

☒ Examiner's attention is drawn to the following co-pending applications, copies of which have been or are being submitted:

Serial No. 10/127,967 filed April 22, 2002

Serial No. 10/127,888 filed April 22, 2002

Serial No. 10/127,938 filed April 22, 2002

□ Other:____

Submission of the above information is not intended as an admission that any item is citable under the statutes or rules to support a rejection, that any item disclosed represents analogous art, or that those skilled in the art would refer to or recognize the pertinence of any reference without the benefit of hindsight, nor should an inference be drawn as to the pertinence of the references based on the order in which they are presented. Submission of this statement should not be taken as an indication that a search has been conducted, or that no better art exists.

It is respectfully requested that the cited information be expressly considered during the prosecution of this application and the references made of record therein.

FEES

☒	**37 CFR 1.97(b):** No fee is believed due in connection with this submission, because the information disclosure statement submitted herewith is satisfies one of the following conditions ("X" indicates satisfaction): ☒ Within three months of the filing date of a national application other than a continued prosecution application under 37 CFR 1.53(d), or □ Within three months of the date of entry into the national stage of an international application as set forth in 37 CFR 1.491 or □ Before the mailing date of a first Office Action on the merits, or □ Before the mailing of a first Office action after the filing of a request for continued examination under 37 CFR 1.114. Although no fee is believed due, if any fee is deemed due in connection with this submission, please charge such fee to Deposit Account 19-1970.
□	**37 CFR 1.97(c):** The information disclosure statement transmitted herewith is being filed after all the above conditions (37 CFR 1.97(b)), but before the mailing date of one of the following conditions: (1) a final action under 37 C.F.R. 1.113 or (2) a notice of allowance under 37 C.F.R. 1.311, or (3) an action that otherwise closes prosecution in the application. This Information Disclosure Statement is accompanied by: □ A Certification (below) as specified by 37 C.F.R. 1.97(e). Although no fee is believed due, if any fee is deemed due in connection with this submission, please charge such fee to Deposit Account 19-1970. OR □ A check in the amount of $180.00 for the fee set forth in 37 C.F.R. 1.17(p) for submission of an information disclosure statement. Please credit any overpayment or charge any underpayment to Deposit Account No. 19-1970.
□	**37 CFR 1.97(d):** This Information Disclosure Statement is being submitted after the period specified in 37 CFR 1.97(c). □ This information Disclosure Statement includes a Certification (below) as specified by 37 C.F.R. 1.97(e) AND □ Applicants hereby requests consideration of the reference(s) disclosed herein. Enclosed is the fee in the amount of $180.00 under 37 C.F.R. 1.17(p). Please credit any overpayment or charge any underpayment to Deposit Account No. 19-1970. Election to pay the fee should not be taken as an indication that applicant(s) cannot execute a certification.

Certification (37 C.F.R. 1.97(e))
(Applicable only if checked)

□ The undersigned certifies that:

□ Each item of information contained in this information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this statement. 37 C.F.R. 1.97(e)(1).

□ A copy of the communication from the foreign patent office is enclosed.

OR

□ No item of information contained in this information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the undersigned after making reasonable inquiry, no item of information contained in this Information Disclosure Statement was known to any individual designated in 37 C.F.R. 1.56(c) more than more than three months prior to the filing of this statement. 37 C.F.R. 1.97(e)(2).

Respectfully submitted,

SHERIDAN ROSS P.C.

By: Douglas Swartz
Douglas W. Swartz
Registration No. 37,739
1560 Broadway, Suite 1200
Denver, Colorado 80202-5141
(303) 863-9700

Date: June 19, 2003

SHEET 1 OF 2

FORM PTO-1449 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT (Use several sheets if necessary)	ATTY. DOCKET NO. 4366-133	SERIAL NO. Not Yet Assigned
	APPLICANT BARDZIL et al.	
	FILING DATE Herewith	GROUP ART

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL		DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE IF APPROP.
	AA	4,644,532	02/17/87	George et al.	370	94	
	AB	5,450,408	09/12/95	Phaal	370	85.13	
	AC	5,557,745	09/17/96	Perlman et al.	395	200.02	
	AD	5,734,824	03/31/98	Choi	395	200.11	
	AE	5,850,397	12/15/98	Raab et al.	370	392	
	AF	5,805,593	09/08/98	Busche	370	396	
	AG	5,881,246	03/09/99	Crawley et al.	395	200.68	
	AH	5,943,317	08/24/99	Brabson et al.	370	238	
	AI	5,966,513	10/12/99	Horikawa et al.	370	200.53	
	AJ	6,256,675	07/03/01	Rabinovich	709	241	
	AK	6,360,255	03/19/02	McCormack et al.	709	221	
	AL	6,405,248	06/11/02	Wood	709	223	
	AM	6,430,612	08/06/02	Iizuka	709	223	
	AN	6,456,306	09/24/02	Chin et al.	345	810	
	AO	2001/0049786	12/06/01	Harrison et al.	713	156	
	AP	2002/0112062	08/15/02	Brown et al.	709	229	
	AQ	2002/0144149	10/03/02	Hanna et al.	713	201	
	AR	2002/0116647	08/22/02	Mont et al.	713	201	
	AS	2002/0161591	10/31/02	Danneels et al.	705	1	
	AT	2002/0087704	07/04/02	Chesnais et al.	709	228	
	AU	2003/0043820	03/06/03	Goringe et al.	370	254	4/22/02

EXAMINER	DATE CONSIDERED

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

SHEET 2 OF 2

FORM PTO-1449 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT (Use several sheets if necessary)	ATTY. DOCKET NO. 4366-133	SERIAL NO. Not Yet Assigned
	APPLICANT BARDZIL et al.	
	FILING DATE Herewith	GROUP ART

	AV	2003/0046427	03/06/03	Goringe et al.	709	242	4/22/02
	AW	10/127,938		Goringe et al.			4/22/02

OTHER ART (Including Author, Title, Date, Pertinent Pages, etc.)

	AX	NET-SNMP, The NET-SNMP Project Home Page, December 13, 2000, 5 pages, http://net-snmp.sourceforge.net
	AY	OpenSSL, The Open Source Toolkit for SSL/TLS, April 17, 2002, 2 pages, http://www.openssl.org
	AZ	Network Working Group, Management Information Base for Network Management of TCP/IP-based Internets: MIB-II, March 1991, 62 pages, http://www.ietf.org/rfc/rfc1213.txt
	BA	Network Working Group, The OSPF NSSA Option, March 1994, 15 pages, http://www.ietf.org/rfc/rfc1587.txt
	BB	Network Working Group, OSPF Version 2, April 1998, 191 pages, http://www.ietf.org/rfc/rfc2328.txt
	BC	Network Working Group, OSPF Version 2 Management Information Base, November 1995, 71 pages, . http://www.ietf.org/rfc/rfc1850.txt
	BD	Network Working Group, RIP Version 2, November 1998, 35 pages, http://www.ierf.org/rfc/rfc2453.txt
	BE	Moy, J., Network Working Group, OSPF Version 2, March 1994, pp. 61-76, 85-95.
	BF	Jason Novotny et al., "An Online Credential Repository for the Grid: MyProxy" from High Performance Distributed Computing, 2001 Proceedings, Berkely, CA (Aug. 2001), pp. 104-111.
	BG	Packet Design CNS, "Route Explorer™ Simplifying Route Analysis", undated, 4 pages.
	BH	Packet Design, Inc., "Route Explorer™ – Reports, Alerts, and Queries", undated, 2 pages.
	BI	Harvey B. Newman, [Henp-net-1] Network Monitoring – Route Explorer (complementary to SNMP?), May 29, 2002, available at http://lists.bnl.gov/pipermail/henp-net-1/2002-May/000156.html, 3 pages.
	BJ	Jim Duffy, "Head in the Clouds," The Edge, May 24, 2002, available at http://www.nwfusion.com/edge/columnists/2002/0520edge2.html, 3 pages.
	BK	Scott Tyler Shafer, "Packet Design Unveils Layer 3 Switch," InfoWorld, May 20, 2002, available at http://www.infoworld.com/article/02/05/20/020520hnestrin_1.html, 2 pages.
	BL	Moy, J., OSPF Version 2 Memorandum to Network Working Group, March 1994, 2 pages.

EXAMINER	DATE CONSIDERED
*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.	

EXHIBIT B

SHEET 2 OF 2

FORM PTO-1449 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT (Use several sheets if necessary)	ATTY. DOCKET NO. 4366-133	SERIAL NO. Not Yet Assigned
	APPLICANT BARDZIL et al.	
	FILING DATE Herewith	GROUP ART

	AV	2003/0046427	03/06/03	Goringe et al.	709	242	4/22/02
	AW	10/127,938		Goringe et al.			4/22/02

OTHER ART (Including Author, Title, Date, Pertinent Pages, etc.)

S.N	AX	NET-SNMP, The NET-SNMP Project Home Page, December 13, 2000, 5 pages, http://net-snmp.sourceforge.net
S.N	AY	OpenSSL, The Open Source Toolkit for SSL/TLS, April 17, 2002, 2 pages, http://www.openssl.org
S.N	AZ	Network Working Group, Management Information Base for Network Management of TCP/IP-based Internets: MIB-II, March 1991, 62 pages, http://www.ietf.org/rfc/rfc1213.txt
S.N	BA	Network Working Group, The OSPF NSSA Option, March 1994, 15 pages, http://www.ietf.org/rfc/rfc1587.txt
S.N	BB	Network Working Group, OSPF Version 2, April 1998, 191 pages, http://www.ietf.org/rfc/rfc2328.txt
S.N	BC	Network Working Group, OSPF Version 2 Management Information Base, November 1995, 71 pages, http://www.ietf.org/rfc/rfc1850.txt
S.N	BD	Network Working Group, RIP Version 2, November 1998, 35 pages, http://www.ierf.org/rfc/rfc2453.txt
S.N	BE	Moy, J., Network Working Group, OSPF Version 2, March 1994, pp. 61-76, 85-95.
S.N	BF	Jason Novotny et al., "An Online Credential Repository for the Grid: MyProxy" from High Performance Distributed Computing, 2001 Proceedings, Berkely, CA (Aug. 2001), pp. 104-111.
S.N	BG	Packet Design CNS, "Route Explorer™ Simplifying Route Analysis", undated, 4 pages.
S.N	BH	Packet Design, Inc., "Route Explorer™ – Reports, Alerts, and Queries", undated, 2 pages.
S.N	BI	Harvey B. Newman, [Henp-net-1] Network Monitoring – Route Explorer (complementary to SNMP?), May 29, 2002, available at http://lists.bnl.gov/pipermail/henp-net-1/2002-May/000156.html, 3 pages.
S.N	BJ	Jim Duffy, "Head in the Clouds," The Edge, May 24, 2002, available at http://www.nwfusion.com/edge/columnists/2002/0520edge2.html, 3 pages.
S.N	BK	Scott Tyler Shafer, "Packet Design Unveils Layer 3 Switch," InfoWorld, May 20, 2002, available at http://www.infoworld.com/article/02/05/20/020520hnestrin_1.html, 2 pages.
S.N	BL	Moy, J., OSPF Version 2 Memorandum to Network Working Group, March 1994, 2 pages.

EXAMINER SARGON NANO	DATE CONSIDERED 6-7-2007

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

EXHIBIT C

Correction

SHEET 2 OF 2

FORM PTO-1449 U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT (Use several sheets if necessary)	ATTY. DOCKET NO. 4366-133	SERIAL NO. Not Yet Assigned
	APPLICANT BARDZIL et al.	
	FILING DATE Herewith	GROUP ART

	AV	2003/0046427	03/06/03	Goringe et al.	709	242	4/22/02
	AW	10/127,938		Goringe et al.			4/22/02

OTHER ART (Including Author, Title, Date, Pertinent Pages, etc.)

S.N	AX	NET-SNMP, The NET-SNMP Project Home Page, December 13, 2000, 5 pages, http://net-snmp.sourceforge.net
S.N	AY	OpenSSL, The Open Source Toolkit for SSL/TLS, April 17, 2002, 2 pages, http://www.openssl.org
S.N	AZ	Network Working Group, Management Information Base for Network Management of TCP/IP-based Internets: MIB-II, March 1991, 62 pages, http://www.ietf.org/rfc/rfc1213.txt
S.N	BA	Network Working Group, The OSPF NSSA Option, March 1994, 15 pages, http://www.ietf.org/rfc/rfc1587.txt
S.N	BB	Network Working Group, OSPF Version 2, April 1998, 191 pages, http://www.ietf.org/rfc/rfc2328.txt
S.N	BC	Network Working Group, OSPF Version 2 Management Information Base, November 1995, 71 pages, http://www.ietf.org/rfc/rfc1850.txt
S.N	BD	Network Working Group, RIP Version 2, November 1998, 35 pages, http://www.ierf.org/rfc/rfc2453.txt
S.N	BE	Moy, J., Network Working Group, OSPF Version 2, March 1994, pp. 61-76, 85-95.
S.N	BF	Jason Novotny et al., "An Online Credential Repository for the Grid: MyProxy" from High Performance Distributed Computing, 2001 Proceedings, Berkely, CA (Aug. 2001), pp. 104-111.
✓ S.N	BG	Packet Design CNS, "Route Explorer™ Simplifying Route Analysis", undated, 4 pages. DATE UNAVAILABLE S.N
✓ S.N	BH	Packet Design, Inc., "Route Explorer™ – Reports, Alerts, and Queries", undated, 2 pages. DATE UNAVAILABLE
S.N	BI	Harvey B. Newman, [Henp-net-1] Network Monitoring – Route Explorer (complementary to SNMP?), May 29, 2002, available at http://lists.bnl.gov/pipermail/henp-net-1/2002-May/000156.html, 3 pages.
S.N	BJ	Jim Duffy, "Head In the Clouds," The Edge, May 24, 2002, available at http://www.nwfusion.com/edge/columnists/2002/0520edge2.html, 3 pages.
S.N	BK	Scott Tyler Shafer, "Packet Design Unveils Layer 3 Switch," InfoWorld, May 20, 2002, available at http://www.infoworld.com/article/02/05/20/020520hnestrin_1.html, 2 pages.
S.N	BL	Moy, J., OSPF Version 2 Memorandum to Network Working Group, March 1994, 2 pages.

EXAMINER SARGON NANO	DATE CONSIDERED 6-7-2007

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

EXHIBIT D

US 20030046427A1

(19) United States
(12) Patent Application Publication
Goringe et al.

(10) Pub. No.: US 2003/0046427 A1
(43) Pub. Date: Mar. 6, 2003

(54) TOPOLOGY DISCOVERY BY PARTITIONING MULTIPLE DISCOVERY TECHNIQUES

(76) Inventors: **Christopher M. Goringe**, Seven Hills (AU); **Muneyb Minhazuddin**, Quakers Hill (AU); **James D. Schreuder**, Summer Hill (AU); **Alex M. Krumm-Heller**, Gladesville (AU)

Correspondence Address:
Douglas W. Swartz
SHERIDAN ROSS P.C.
Suite 1200
1560 Broadway
Denver, CO 80202-5141 (US)

(21) Appl. No.: **10/127,888**

(22) Filed: **Apr. 22, 2002**

Related U.S. Application Data

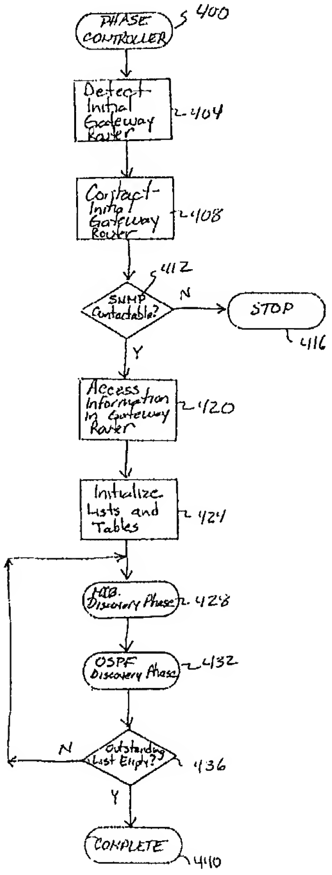
(60) Provisional application No. 60/317,719, filed on Sep. 6, 2001. Provisional application No. 60/347,050, filed on Jan. 8, 2002.

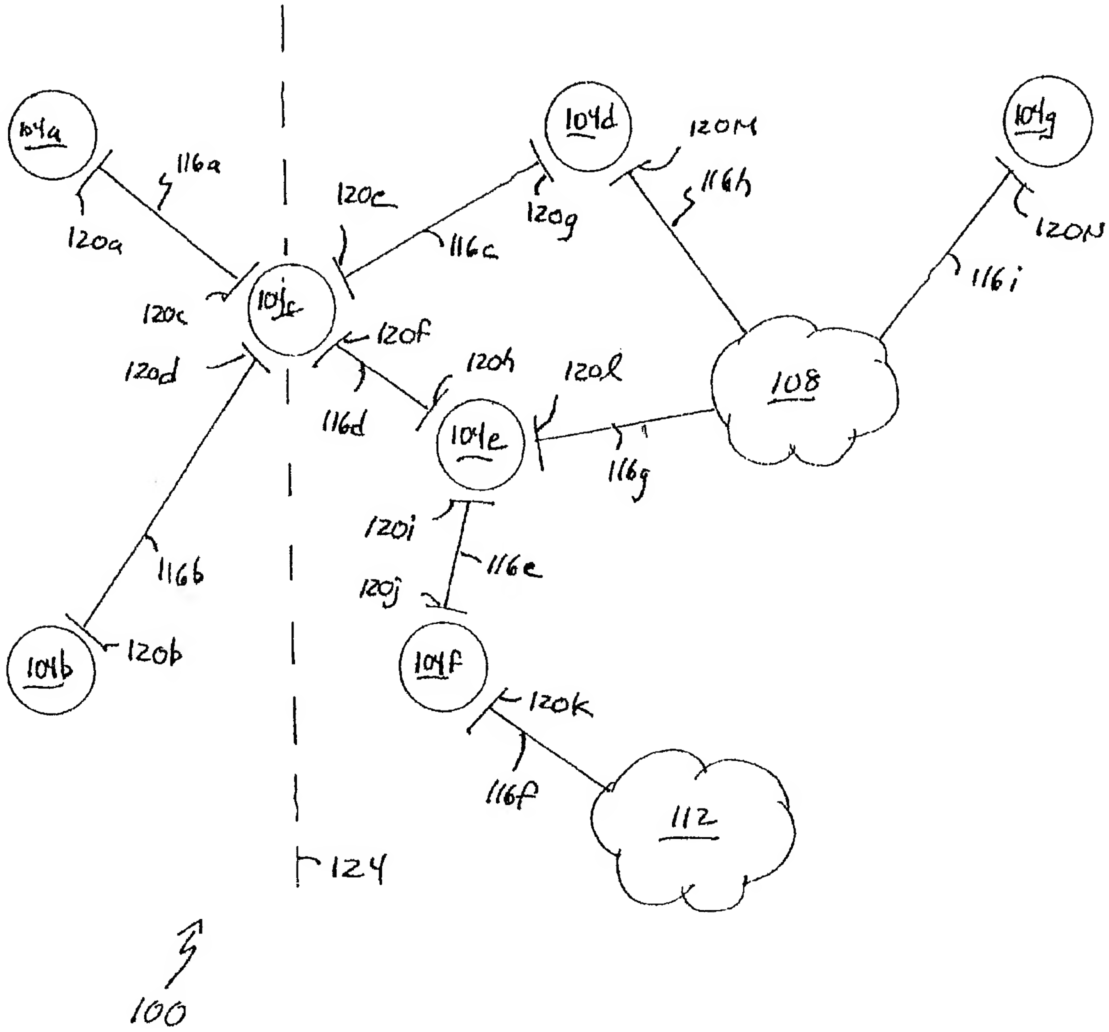
Publication Classification

(51) Int. Cl.^{7} **G06F 15/173**
(52) U.S. Cl. **709/242**; 709/224

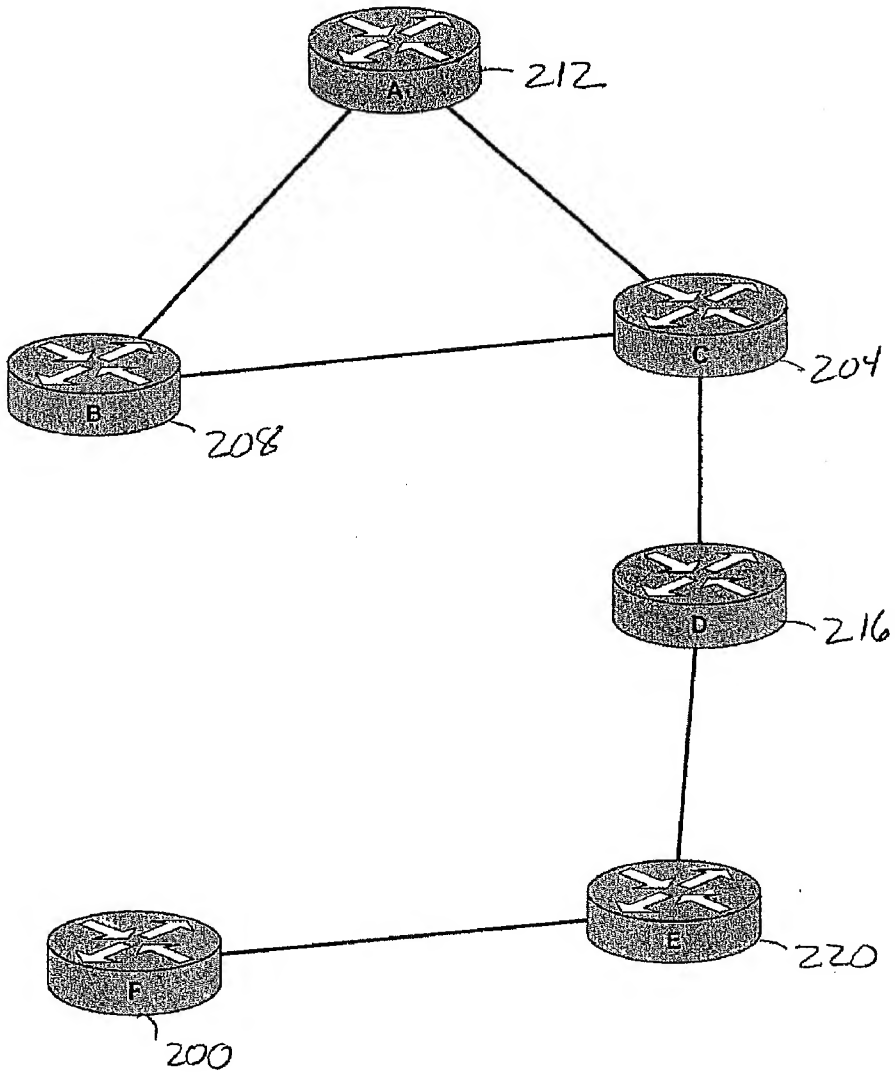
(57) **ABSTRACT**

A system for discovering a topology of a distributed processing network that includes a first topology discovery agent 308 configured to contact a first set of routers to obtain a first type of information stored in each router in the first set of routers; a second topology discovery agent 312 and/or 316 configured to contact a second set of routers to obtain a second type of information stored in each router in the second set of routers, and a phase controller 304 configured to select between the first and second topology discovery agents. The first and second sets of routers are different, and the first and second types of information are different. In one configuration, the first type of information is defined by a network management protocol, and the second type of information is defined by a routing protocol.





Prior Art
FIG. 1



PRIOR ART

Figure 2

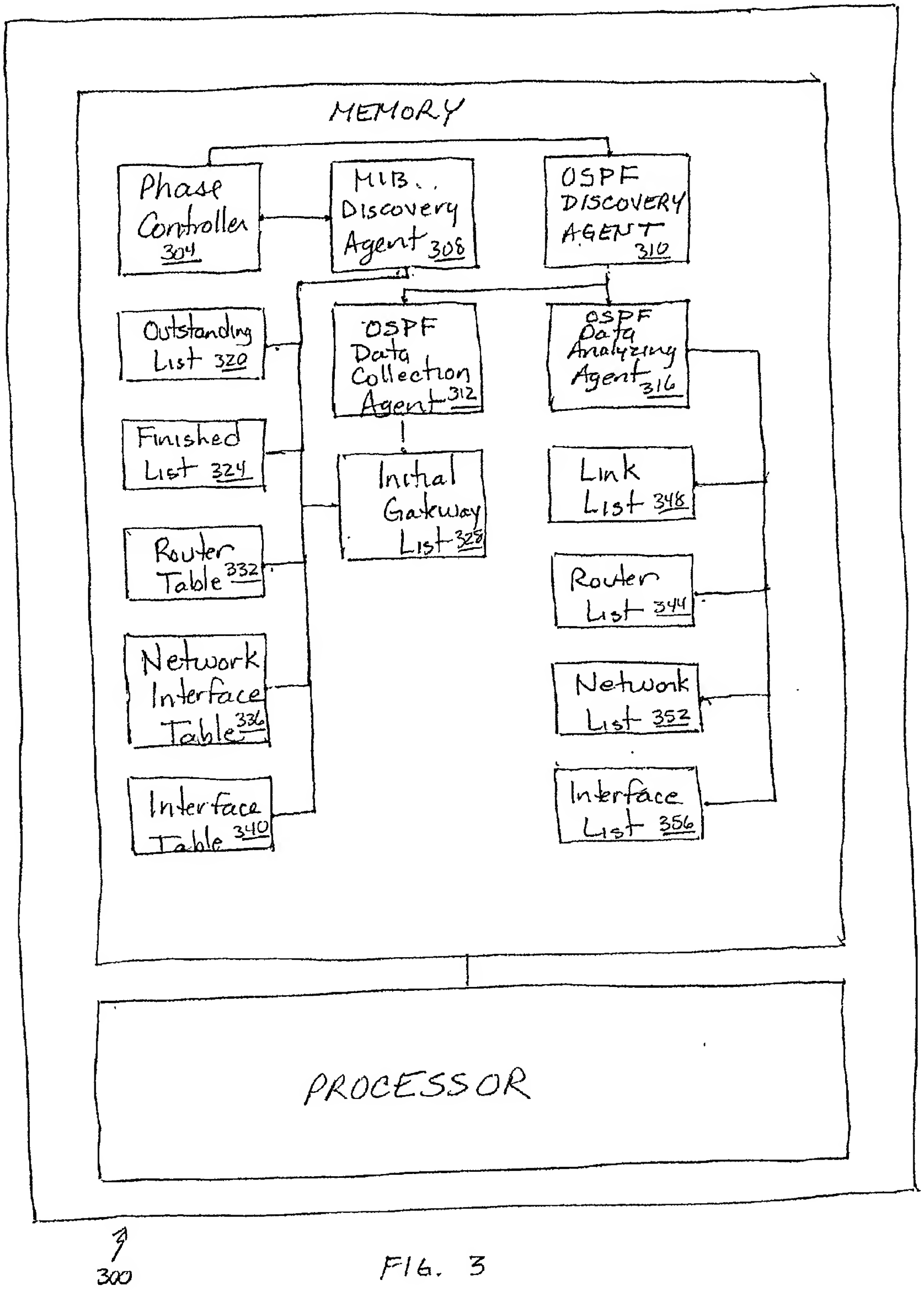

MEMORY
Phase Controller 304
MIB Discovery Agent 308
OSPF DISCOVERY AGENT 310
Outstanding List 320
OSPF Data Collection Agent 312
OSPF Data Analyzing Agent 316
Finished List 324
Initial Gateway List 328
Link List 348
Router Table 332
Router List 344
Network Interface Table 336
Network List 352
Interface Table 340
Interface List 356
PROCESSOR
300

FIG. 3

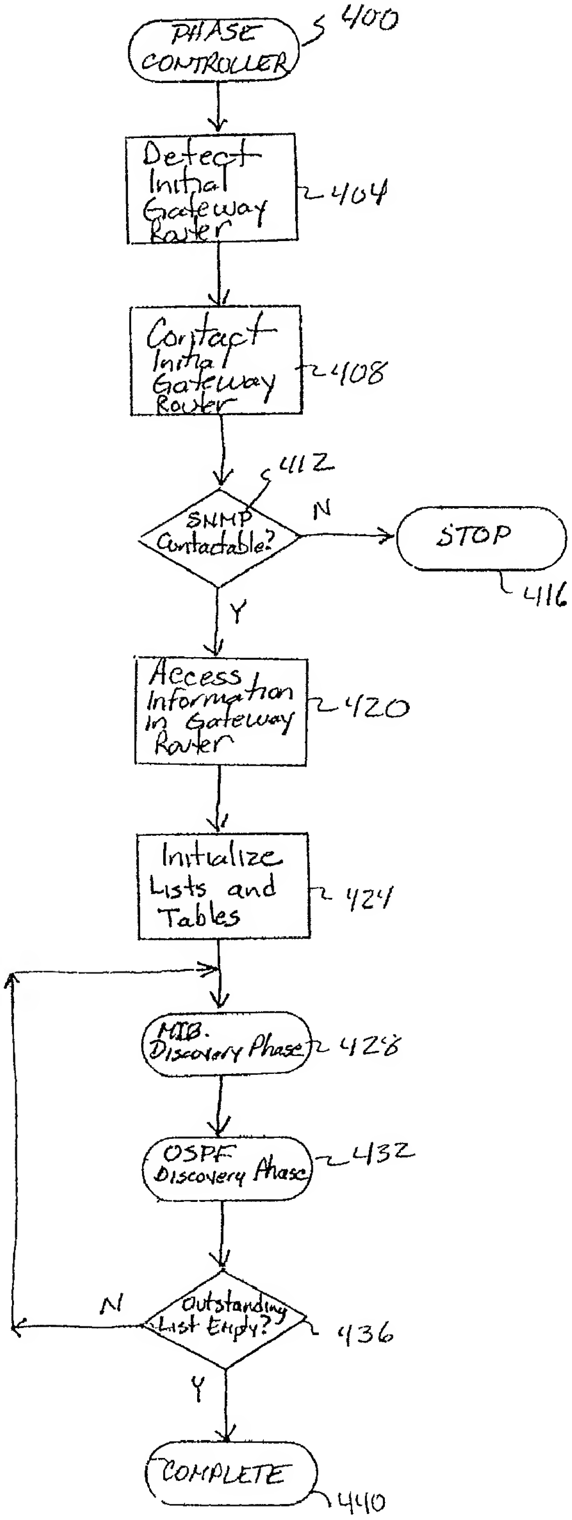

PHASE CONTROLLER
400
Detect Initial Gateway Router
404
Contact Initial Gateway Router
408
412
SNMP Contactable?
N
STOP
416
Y
Access Information in Gateway Router
420
Initialize Lists and Tables
424
MIB Discovery Phase
428
OSPF Discovery Phase
432
N
Outstanding List Empty?
436
Y
COMPLETE
440

FIG. 4

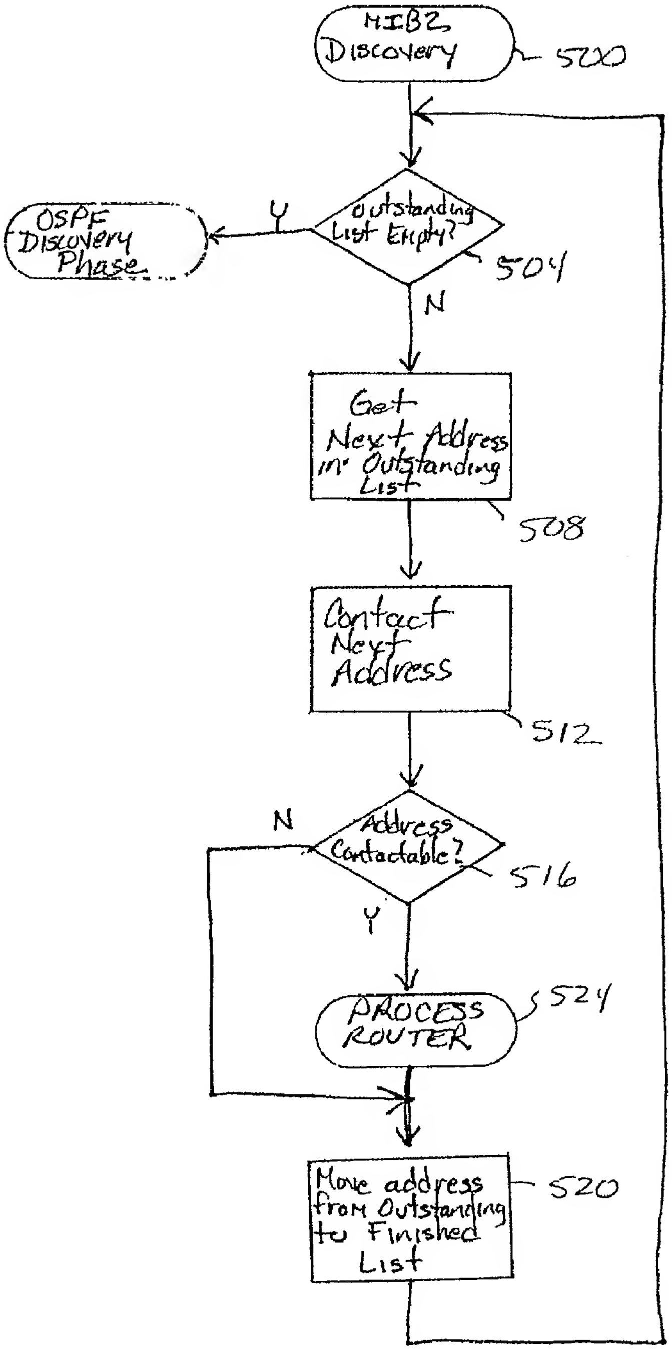

MIB2 Discovery
500
Outstanding List Empty?
504
Y
OSPF Discovery Phase
N
Get Next Address in Outstanding List
508
Contact Next Address
512
Address Contactable?
516
N
Y
PROCESS ROUTER
524
Move address from Outstanding to Finished List
520

FIG. 5

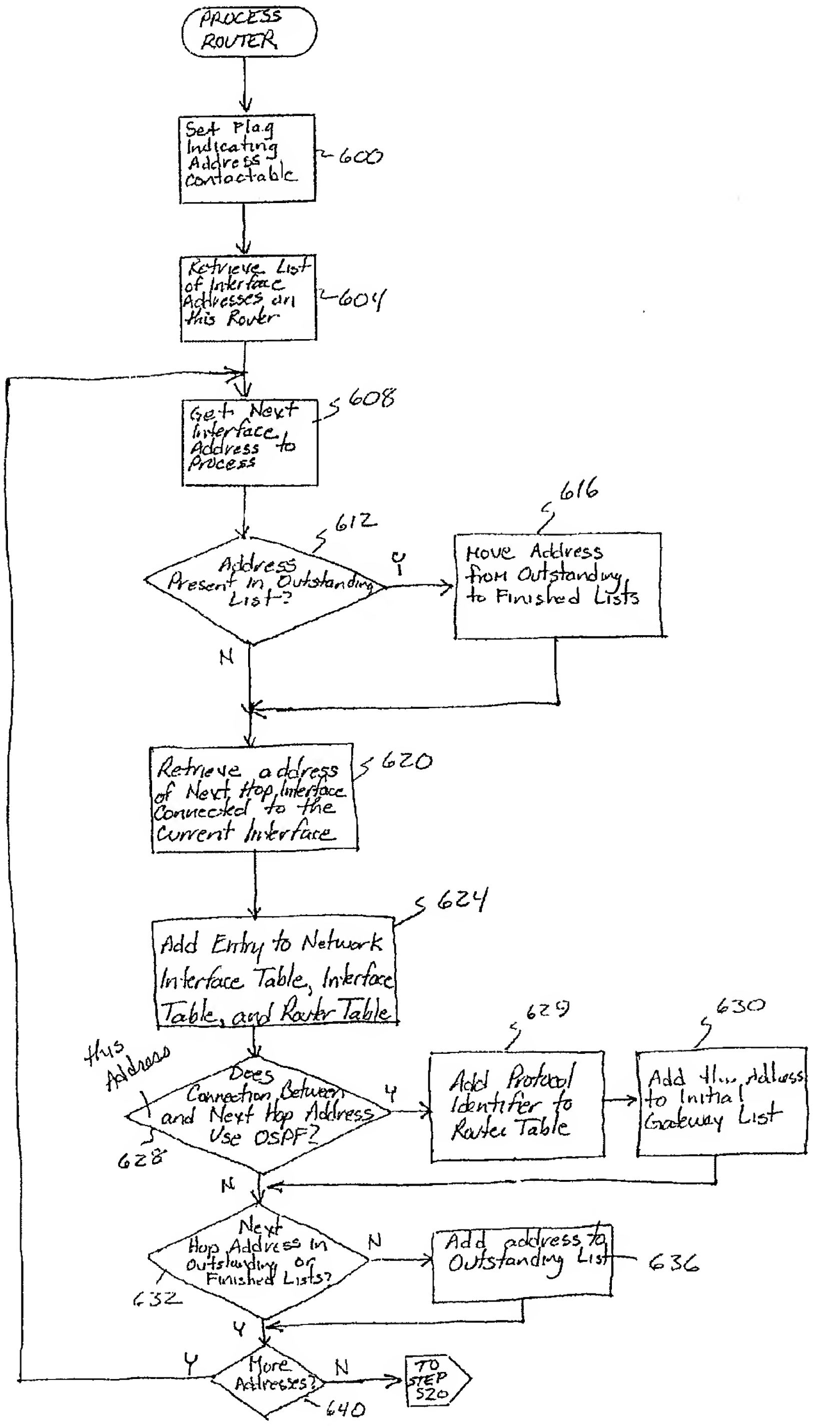

PROCESS ROUTER
Set Flag Indicating Address Contactable
600
Retrieve List of Interface Addresses on this Router
604
Get Next Interface Address to Process
608
Address Present in Outstanding List?
612
Y
N
Move Address from Outstanding to Finished Lists
616
Retrieve address of Next Hop Interface Connected to the Current Interface
620
Add Entry to Network Interface Table, Interface Table, and Router Table
624
Does Connection Between this Address and Next Hop Address Use OSPF?
628
Y
N
Add Protocol Identifier to Router Table
629
Add this Address to Initial Gateway List
630
Next Hop Address in Outstanding or Finished Lists?
632
N
Y
Add address to Outstanding List
636
More Addresses?
640
Y
N
TO STEP 520

FIG. 6

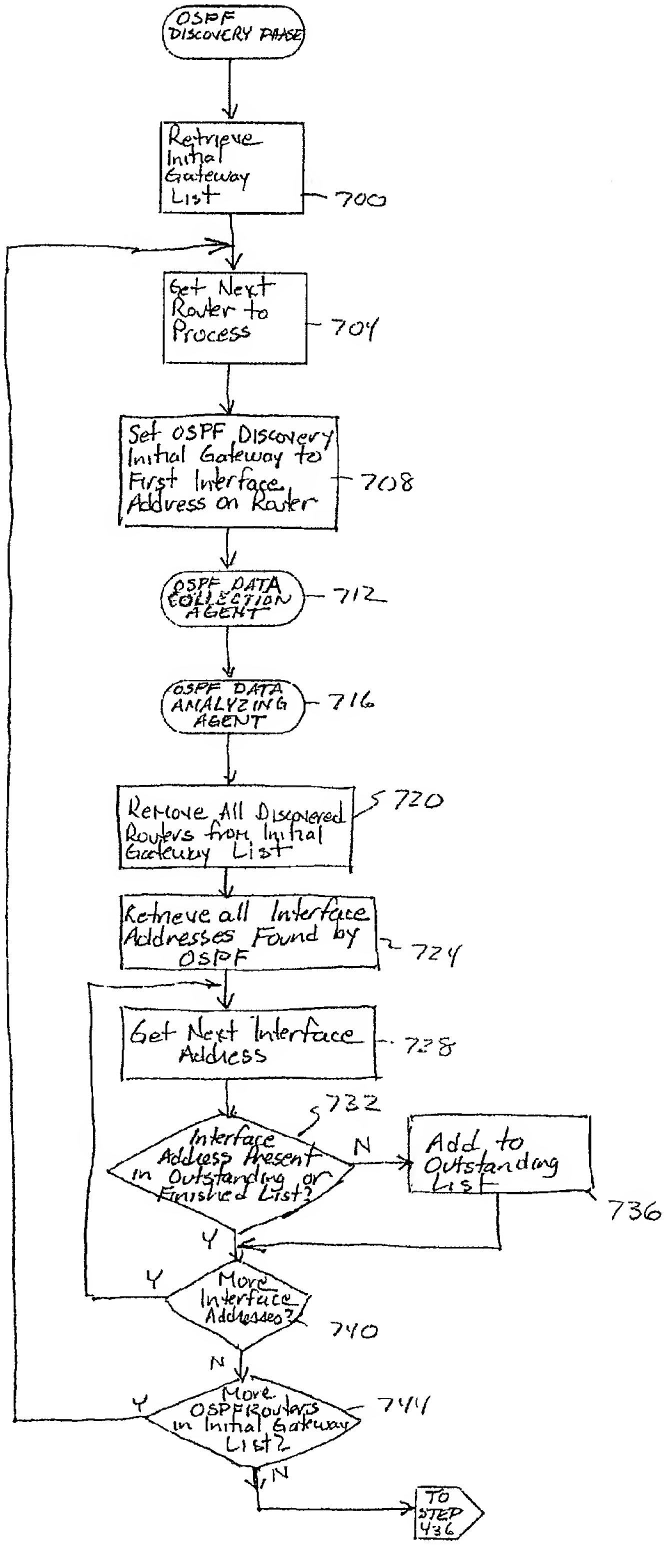

OSPF DISCOVERY PHASE
Retrieve Initial Gateway List
700
Get Next Router to Process
704
Set OSPF Discovery Initial Gateway to First Interface Address on Router
708
OSPF DATA COLLECTION AGENT
712
OSPF DATA ANALYZING AGENT
716
Remove All Discovered Routers from Initial Gateway List
720
Retrieve all Interface Addresses Found by OSPF
724
Get Next Interface Address
728
732
Interface Address Present in Outstanding or Finished List?
N
Add to Outstanding List
736
Y
Y
More Interface Addresses?
740
N
Y
More OSPF Routers in Initial Gateway List?
744
N
TO STEP 436

FIG. 7

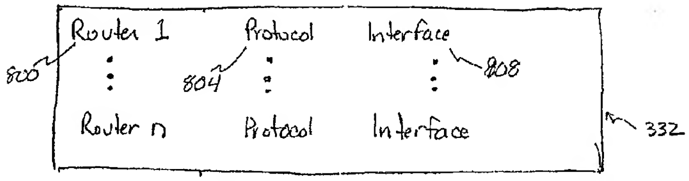


FIG 8

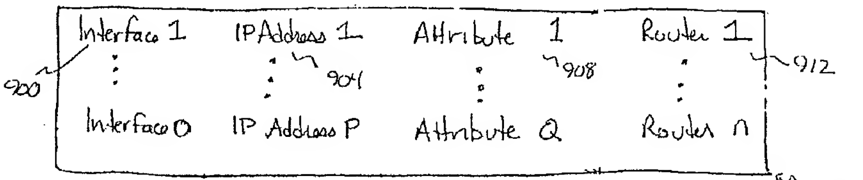


FIG. 9

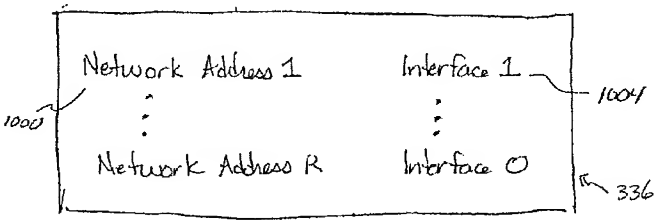


FIG. 10

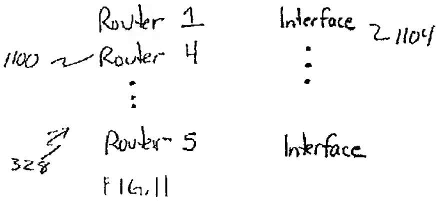


FIG. 11

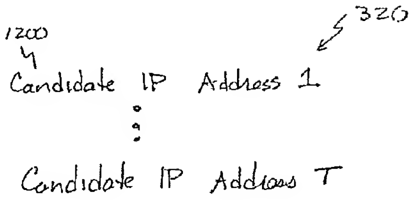


FIG. 12

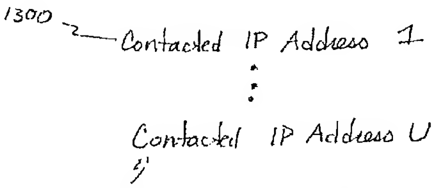


FIG. 13

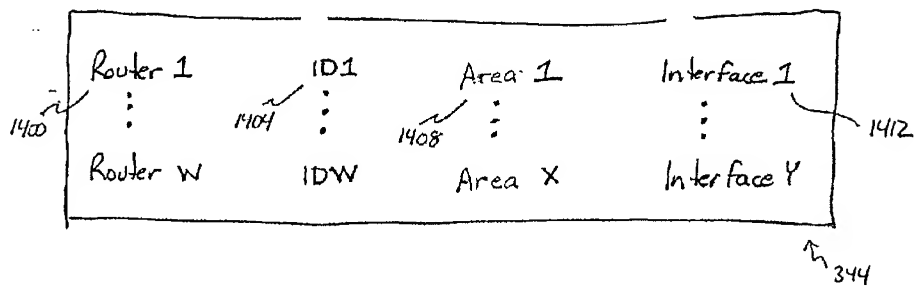


FIG. 14

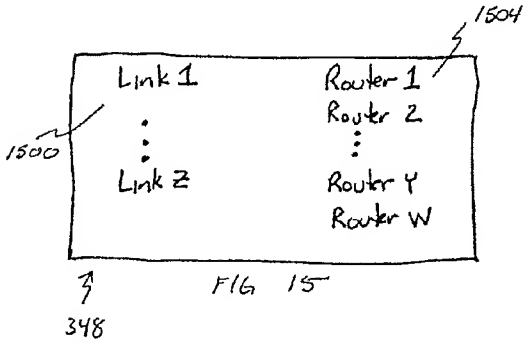


FIG 15

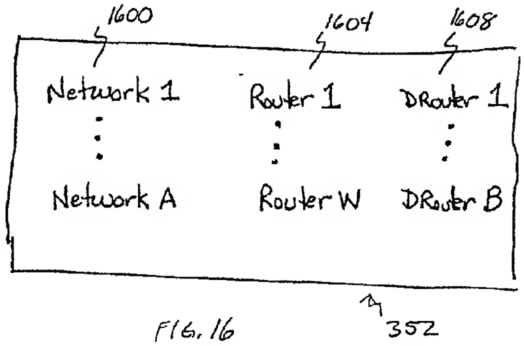


FIG. 16

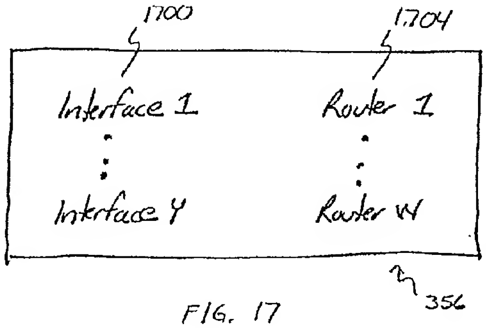
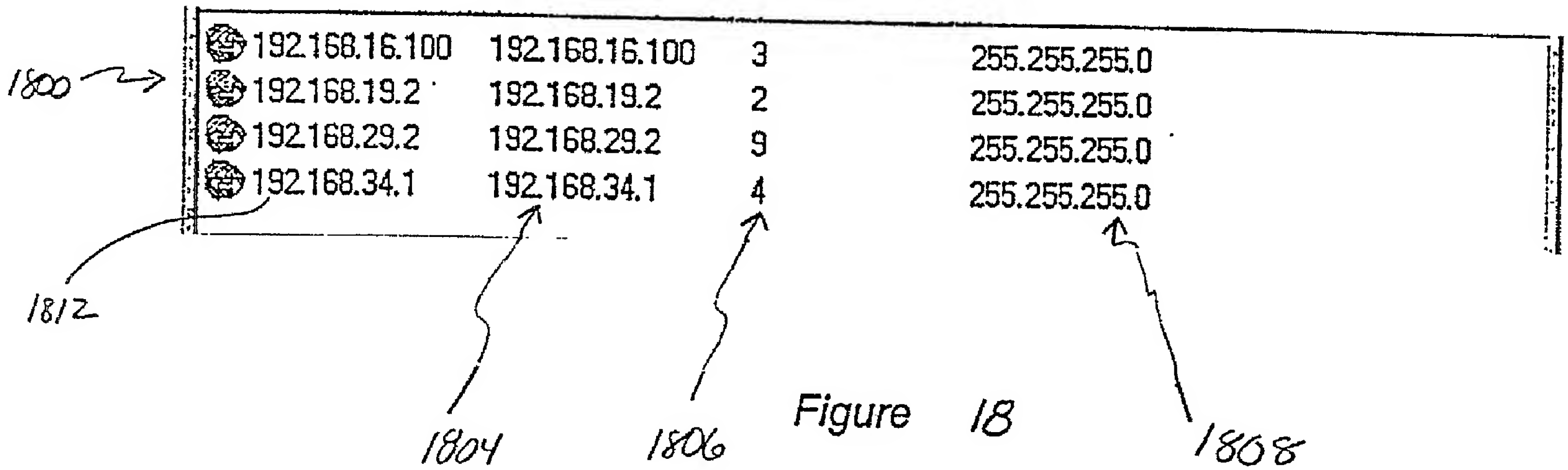
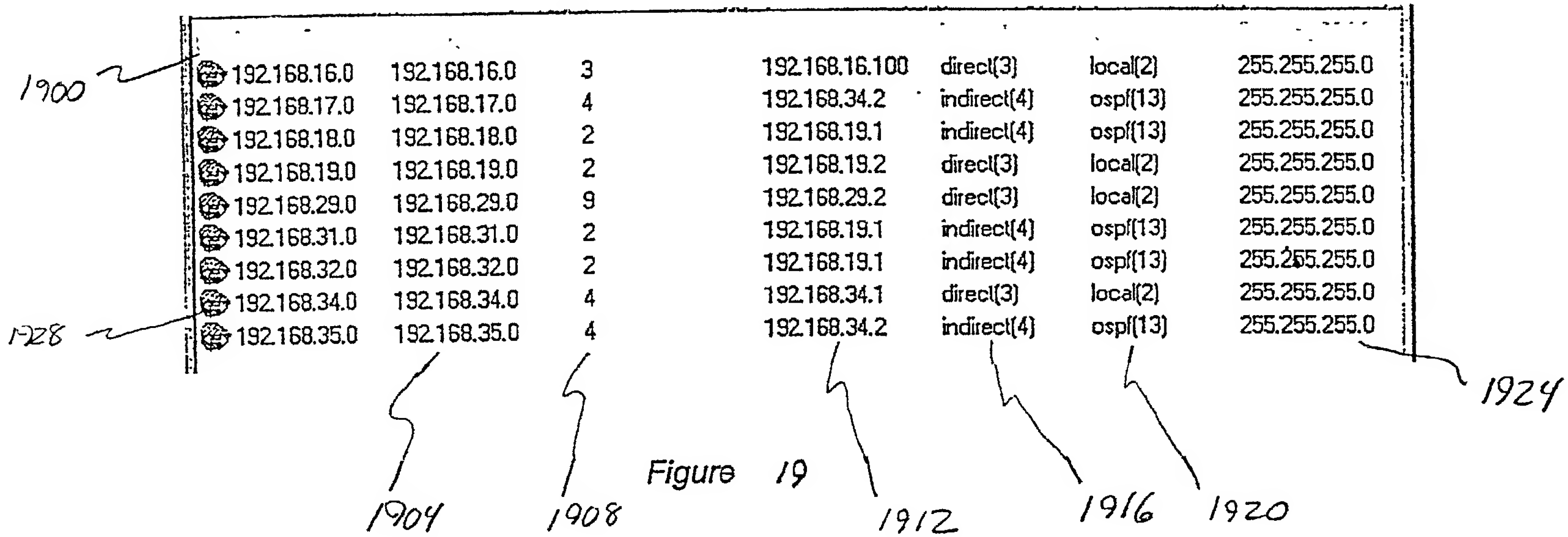


FIG. 17



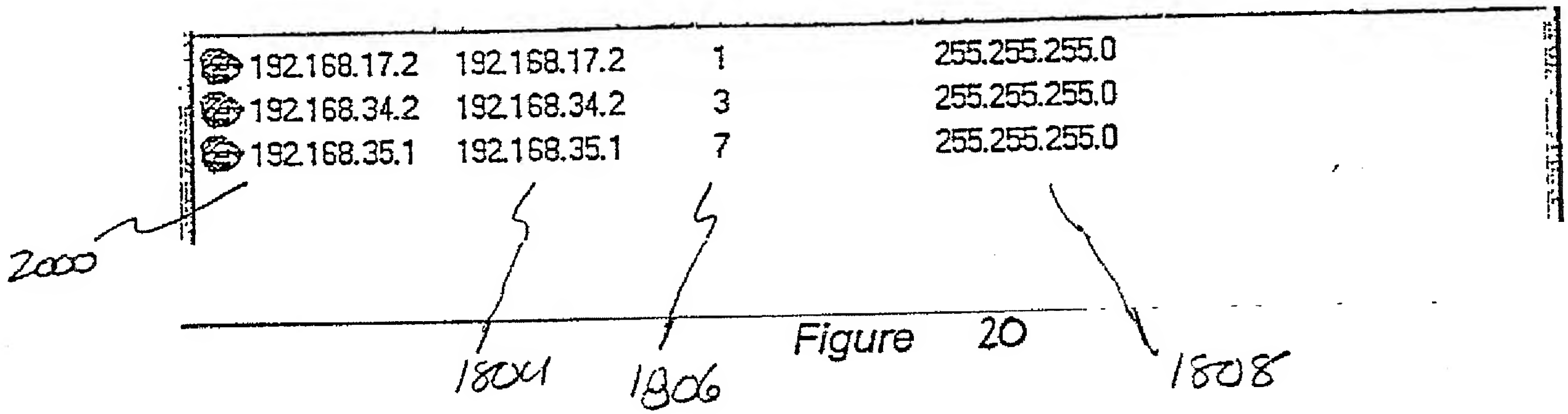
192.168.16.100	192.168.16.100	3	255.255.255.0
192.168.19.2	192.168.19.2	2	255.255.255.0
192.168.29.2	192.168.29.2	9	255.255.255.0
192.168.34.1	192.168.34.1	4	255.255.255.0

Figure 18



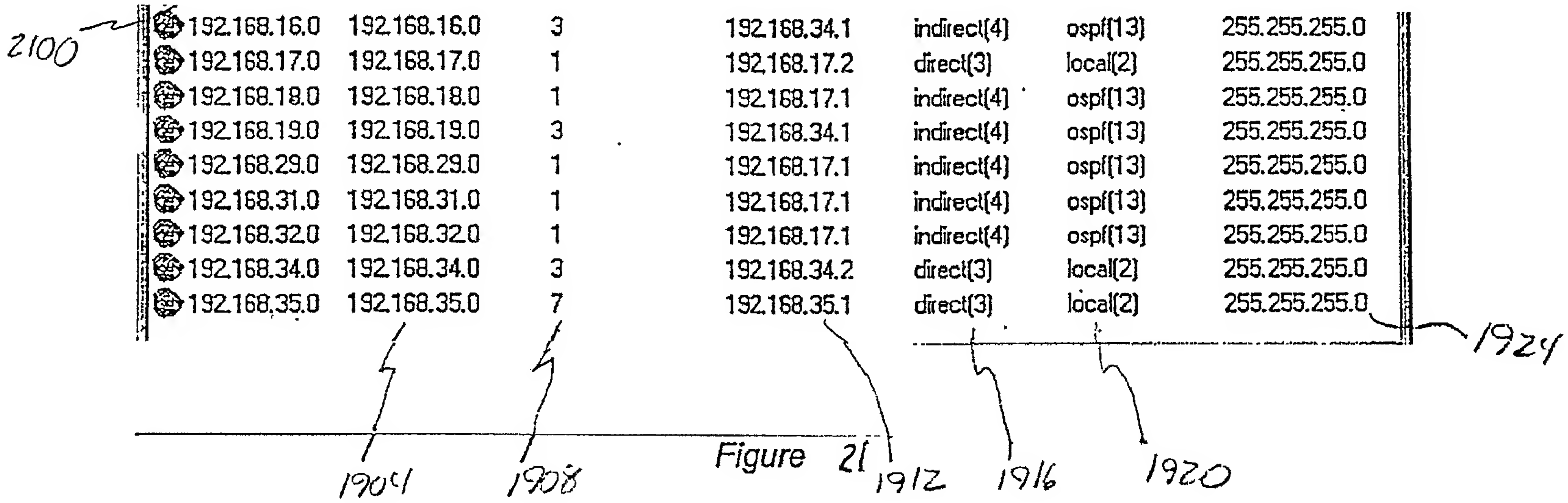
192.168.16.0	192.168.16.0	3	192.168.16.100	direct(3)	local(2)	255.255.255.0
192.168.17.0	192.168.17.0	4	192.168.34.2	indirect(4)	ospf(13)	255.255.255.0
192.168.18.0	192.168.18.0	2	192.168.19.1	indirect(4)	ospf(13)	255.255.255.0
192.168.19.0	192.168.19.0	2	192.168.19.2	direct(3)	local(2)	255.255.255.0
192.168.29.0	192.168.29.0	9	192.168.29.2	direct(3)	local(2)	255.255.255.0
192.168.31.0	192.168.31.0	2	192.168.19.1	indirect(4)	ospf(13)	255.255.255.0
192.168.32.0	192.168.32.0	2	192.168.19.1	indirect(4)	ospf(13)	255.255.255.0
192.168.34.0	192.168.34.0	4	192.168.34.1	direct(3)	local(2)	255.255.255.0
192.168.35.0	192.168.35.0	4	192.168.34.2	indirect(4)	ospf(13)	255.255.255.0

Figure 19



192.168.17.2	192.168.17.2	1	255.255.255.0
192.168.34.2	192.168.34.2	3	255.255.255.0
192.168.35.1	192.168.35.1	7	255.255.255.0

Figure 20



192.168.16.0	192.168.16.0	3	192.168.34.1	indirect(4)	ospf(13)	255.255.255.0
192.168.17.0	192.168.17.0	1	192.168.17.2	direct(3)	local(2)	255.255.255.0
192.168.18.0	192.168.18.0	1	192.168.17.1	indirect(4)	ospf(13)	255.255.255.0
192.168.19.0	192.168.19.0	3	192.168.34.1	indirect(4)	ospf(13)	255.255.255.0
192.168.29.0	192.168.29.0	1	192.168.17.1	indirect(4)	ospf(13)	255.255.255.0
192.168.31.0	192.168.31.0	1	192.168.17.1	indirect(4)	ospf(13)	255.255.255.0
192.168.32.0	192.168.32.0	1	192.168.17.1	indirect(4)	ospf(13)	255.255.255.0
192.168.34.0	192.168.34.0	3	192.168.34.2	direct(3)	local(2)	255.255.255.0
192.168.35.0	192.168.35.0	7	192.168.35.1	direct(3)	local(2)	255.255.255.0

Figure 21

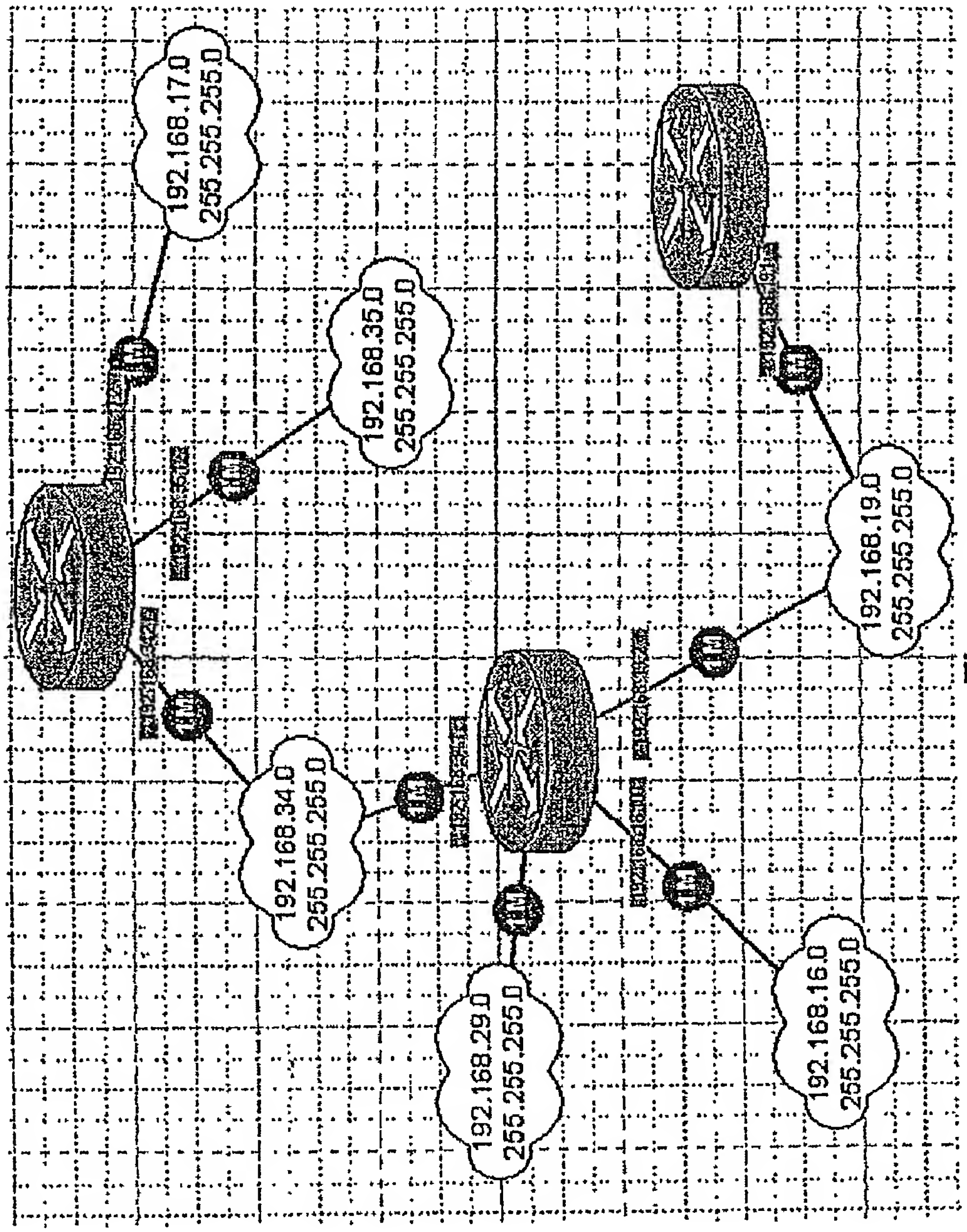

192.168.17.0
255.255.255.0
192.168.35.0
255.255.255.0
192.168.34.0
255.255.255.0
192.168.19.0
255.255.255.0
192.168.29.0
255.255.255.0
192.168.16.0
255.255.255.0

Figure . 22

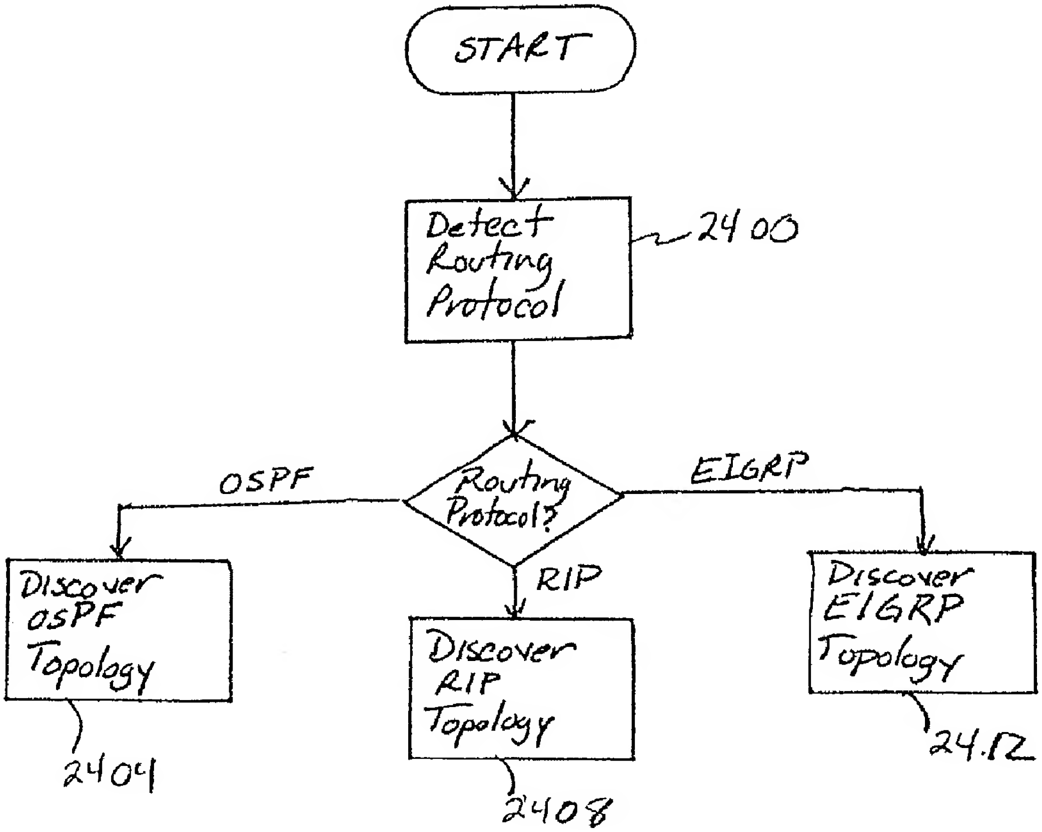

START
Detect Routing Protocol
2400
Routing Protocol?
OSPF
RIP
EIGRP
Discover OSPF Topology
2404
Discover RIP Topology
2408
Discover EIGRP Topology
2412

Fig. 23

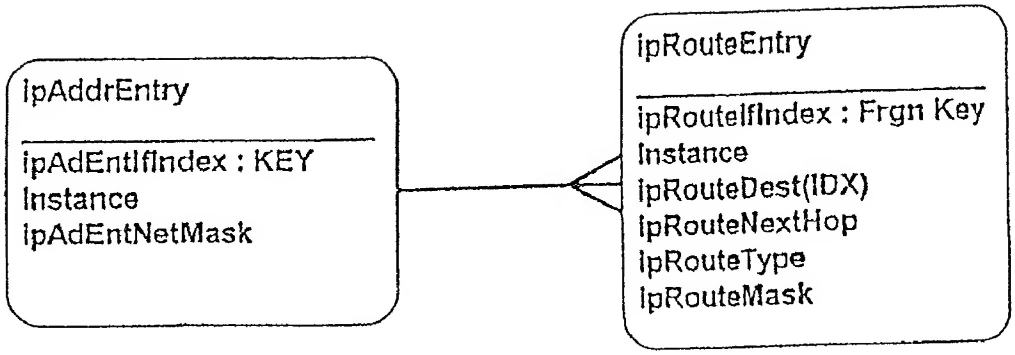

ipAddrEntry
ipAdEntIfIndex : KEY
Instance
ipAdEntNetMask
ipRouteEntry
ipRouteIfIndex : Frgn Key
Instance
ipRouteDest(IDX)
ipRouteNextHop
ipRouteType
ipRouteMask

Figure 24

TOPOLOGY DISCOVERY BY PARTITIONING MULTIPLE DISCOVERY TECHNIQUES

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims priority from U.S. Provisional Application Serial No. 60/317,719, filed Sep. 6, 2001, of the same title, to Goringe, et al., and from U.S. Provisional Application Serial No. 60/347,050 filed Jan. 8, 2002, entitled "Topology Discovery by Partitioning Multiple Discovery Techniques," to Goringe, et. al., each of which is incorporated herein by this reference.

FIELD OF THE INVENTION

[0002] The present invention relates generally to networks and specifically to methods and devices for determining network or routing topology.

BACKGROUND OF THE INVENTION

[0003] Distributed processing networks are gaining increasing importance in our information-based society. **FIG. 1** depicts a network topology of a simple data network. The network **100** comprises a plurality of routers **104***a-g*, a transit network **108**, and a stub network **112**, all interconnected by links **116***a-i*. As will be appreciated, a router is a device connecting two or more networks that routes incoming data or packets to an appropriate network/node; a transit network is a network containing more than router; a stub network is a network that is not configured to transit packets through the network from one router to another; and a link is a communication channel between two or more nodes. Each of the routers is typically attached to a link via one or more interfaces, such as interfaces **120***a-n*. The simple network of **FIG. 1** is divided into two protocol regions with the dashed line **124** being the boundary between the two regions. Router **104***c* is located on the boundary **124** and is referred to herein as an area border router while the other routers **104***a-b* and *d-g* are not area border routers. One or more protocol regions are often autonomous systems. An autonomous system is a collection of networks controlled by a single administrative authority.

[0004] In a packet-switched network, the technique used to route a packet through interconnected networks depends on the routing protocol. Most protocols fall into one of two categories, distance-vector algorithms (which determine the distance between source and destination nodes by calculating the number of router hops a packet traverses en route from the source network to the destination network) and link-state algorithms (which use link state advertisement or LSA (containing the names and various cost metrics of a router's neighbors) to keep routers informed about links in the network). Rather than storing actual paths (which is the case with distance-vector algorithms), link-state algorithms store the information needed to generate such paths. Examples of router protocols using distance-vector algorithms include RIP and RIP-2 and using link-state algorithms include Open Shortest Path First or OSPF, OSI's IS-IS, EIGRP, and Netware's Link Service's Protocol (NLSP).

[0005] Routers and other network components are typically managed using a network management system. Network management systems perform network maintenance, identify possible security problems in the network, locate equipment, module, subassembly, and card failures, locate circuit outages, monitor levels of performance (e.g., bit error rates or BERs, loss of synchronization, etc.) and permit rapid and accurate quantification of network usage and traffic levels. Examples of network management systems used for performing the foregoing tasks include Hewlett-Packard's OpenView™, IBM's Netview™, and Digital Equipment Corporation's Enterprise Management Architecture or EMA™.

[0006] For optimal operation of network management systems, an accurate, detailed map of the network or OSI layer 3 topologies is commonly required. Such a map not only facilitates operation of the network management system but also permits newly attached hosts to be properly located and configured for the network (to avoid adversely impacting network performance) and existing hosts to be properly located for the newly attached host. In common practice, a detailed map of the network's topology is, in whole or part, unavailable to network management personnel. This can be due to poor record keeping, the sheer size and complexity of some networks, and the lack of central management of a network, such as where a network includes a number of autonomous systems or enterprises.

[0007] Simple Network Management Protocol or SNMP algorithms for discovering automatically network layer topology are used in many network management tools. The SNMP algorithms can take several approaches. In one approach known as the "hop-by-hop" approach, the algorithm accesses standard routing SNMP-Management Information Base or MIB information in each router on a hop-by-hop basis. As used herein, a "hop" refers to a portion of a route that has no intermediate nodes and "MIB" is the set of managed objects or variables that can be managed as defined by SNMP. MIB objects or variables are typically defined by the set of rules known as Structure of Management Information or SMI. As will be appreciated, MIB information is stored in the memory of any SNMP router. In another approach, vendor-specific proprietary algorithms are used to generate the topology. An example of such a solution is CDP™ by Cisco Systems. Such proprietary algorithms typically rely on the vendor-specific extensions to the standard SNMP MIBs that are generally not useful in a multi-vendor network.

[0008] SNMP network topology discovery algorithms are typically unable to ascertain Layer 3 topology when the routers in a network support differing routing protocols and/or are uncontactable. A router can be uncontactable for a variety of reasons including the use of improper credentials, a down state of the contacted interface, an inaccessible or nonexistent SNMP agent in the router, etc. This problem is illustrated by **FIG. 2**. Referring to **FIG. 2**, router **200** supports Routing Information Protocol or RIP and is SNMP contactable, routers **204, 208**, and **212** support the Open Shortest Path First or OSPF protocol and are SNMP contactable, router **216** supports OSPF but is not SNMP contactable, and finally router **220** supports OSPF and RIP and is SNMP contactable. If the topology discovery algorithm initially contacts router **200**, it will, by the hop-by-hop approach, be able to discover routers **200, 220** and **216**. When the algorithm contacts router **216**, which is not SNMP contactable, the algorithm will be unable to discover routers **204, 208**, and **212**. This is so because the algorithm will be unable to access the MIB information in router **216**, thereby preventing the algorithm from learning of the existence of these routers.

SUMMARY OF THE INVENTION

[0009] These and other needs are addressed by the various embodiments and configurations of the present invention. Generally, the architecture of the present invention uses multiple topology discovery techniques to discover network topology. The differing discovery techniques can be network management protocol- and/or routing protocol-specific. The use of a network management protocol-specific discovery algorithm can make the algorithm discovery agnostic (or insensitive).

[0010] In one embodiment that is particularly useful for an enterprise network or autonomous system, a hop-by-hop discovery algorithm, such as an SNMP topology discovery technique, is combined with one or more other discovery algorithms, such as an OSPF discovery algorithm, a discovery algorithm based on proprietary standards, and a vendor-specific topology discovery algorithm, to perform topology discovery. The topology discovery mechanism is typically designed to be insensitive to routing protocol. This is made possible by the hop-by-hop approach, where each router is contacted to find out the other entities in the network known to the contacted router. Each router is contacted in turn to build a database regarding the various network entities and their topology connections.

[0011] Whenever a router is discovered by the hop-by-hop method, it is queried to identify which protocols have been used to define its route table. If these protocols have specific discovery algorithms associated with them, the corresponding discovery algorithm is run. Any additional data found is added to a network model, and any additional routers discovered are made available to the hop-by-hop algorithm for further exploration. Routers discovered using the alternative discovery algorithm(s) can be used to "jump over" uncontactable routers such as router **216** in **FIG. 2**. Referring to **FIG. 2**, using an OSPF topology discovery algorithm in addition to an SNMP discovery algorithm would disclose OSPF area information on any of routers **204**, **208**, **212**, and **216**. The area information would identify that the uncontactable router **216** exists, but no other information can then be deduced.

[0012] In another embodiment, the architecture uses a number of discovery agents to discover heterogeneous networks executing a variety of different routing protocols. Each discovery agent is configured to interact with router information defined by one or more of the routing protocols. The various discovery agents can be operated in discrete phases, in parallel, or on a router-by-router basis.

[0013] The architecture of the present invention can have a number of advantages. For example, the use of differing discovery techniques can locate network components even though the network components support differing protocols. This advantage can be illustrated with reference to the configuration of the architecture using both MIB and OSPF discovery techniques. As used herein, "MIB" is considered as referring to all versions of the Management Information Base, "SNMP" to all versions of the Simple Network Management Protocol, and "OSPF" to all versions of the OSPF protocol. The advantage of MIB discovery is that it will discover an IP network topology regardless of what routing protocols are present. The disadvantage of this approach is that the MIB discovery agent must visit each router in the network to discover the entire network. If a router is not contactable via SNMP as discussed above with reference to **FIG. 2**, the MIB discovery agent will discover only a subset of the network. Using OSPF discovery alone will only discover the parts of a network that are executing the OSPF routing protocol. OSPF discovery techniques can, however, quickly identify large areas of a network by contacting only a small number of routers (e.g., area border routers), thereby insulating such techniques from being thwarted by an uncontactable router. Thus, OSPF discovery may find routers that MIB discovery could not, because OSPF can determine a network topology without having to visit each router. In particular, OSPF may discover part of a network that is unreachable due to an uncontactable router blocking MIB discovery as noted above with reference to **FIG. 2**. Using two discovery techniques together allows MIB discovery to find all contactable and uncontactable routers in the network regardless of the routing protocols present in the network. Parts of the network that are not contactable can often be discovered by using OSPF discovery to "hop" over an uncontactable router for later discovery by MIB discovery.

[0014] These and other advantages will be apparent from the disclosure of the invention(s) contained herein.

[0015] The above-described embodiments and configurations are neither complete nor exhaustive. As will be appreciated, other embodiments of the invention are possible utilizing, alone or in combination, one or more of the features set forth above or described in detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] **FIG. 1** depicts a simple network topology according to the prior art;

[0017] **FIG. 2** depicts another simple network topology according to the prior art;

[0018] **FIG. 3** is a block diagram depicting a computational architecture according to an embodiment of the present invention;

[0019] **FIG. 4** is a flowchart depicting the operation of the phase controller;

[0020] **FIG. 5** is a flowchart depicting the operation of the MIB discovery agent;

[0021] **FIG. 6** is another flowchart depicting the operation of the MIB discovery agent;

[0022] **FIG. 7** is a flowchart depicting the operation of the OSPF discovery agent;

[0023] **FIG. 8** is a router table output by the MIB discovery agent;

[0024] **FIG. 9** is an interface table output by the MIB discovery agent;

[0025] **FIG. 10** is a network interface table output by the MIB discovery agent;

[0026] **FIG. 11** is an initial gateway list maintained by the phase controller;

[0027] **FIG. 12** is an outstanding list maintained by the phase controller;

[0028] **FIG. 13** is a finished list maintained by the phase controller;

[0029] **FIG. 14** is a router list output by the OSPF discovery agent;

[0030] **FIG. 15** is a link list output by the OSPF discovery agent;

[0031] **FIG. 16** is a network list output by the OSPF discovery agent;

[0032] **FIG. 17** is an interface list output by the OSPF discovery agent;

[0033] **FIG. 18** is a typical ipAddrTable table;

[0034] **FIG. 19** is a typical ipRouteEntry table;

[0035] **FIG. 20** is a typical ipAddr table;

[0036] **FIG. 21** is a typical ipRoute table;

[0037] **FIG. 22** is a network topology based on selected entries in the ipAddrTable, ipRouteEntry, and ipRoute tables of FIGS. **18**-**19** and **21**;

[0038] **FIG. 23** is a flowchart depicting a multi-routing protocol architecture according to another embodiment of the present invention; and

[0039] **FIG. 24** depicts the relationship between the ipAddr Entry and ipRouteEntry tables.

DETAILED DESCRIPTION

[0040] To discover the topology of a network with multiple routing protocols present, a phased discovery approach is employed. There are three main phases required to discover a network topology. In the initial gateway detection phase, the architecture contacts a seed router to initiate the discovery process. In the MIB or MIB2 discovery phase, the architecture contacts each of the routers in the enterprise network to download selected MIB information in the routers. In the OSPF discovery phase, the architecture contacts routers supporting OSPF to download link state advertisements from the link state advertisement database in the OSPF routers.

[0041] Before discussing the operation of the data collection and analyzing agent **208**, it is important to understand certain features of many routing protocols. A router can be identified by a unique router ID in the case of certain protocols, and associated with a unique area ID. A router typically does not itself have an IP address. An interface is a logical device belonging to a host such as a router than can be the attachment point of a link. Typically, an interface will have zero or one IP address and belong to a network. The interface will normally have an interface number and a network mask. A link contains two or more bindings of a source interface and a metric or cost. It is templated by the metric representation which is specific to the routing protocol and represents the cost for a packet to leave an interface. A link is typically associated with a cost metric and a routing protocol identifier. A network object represents a data network or subnet. It has an address and a mask and represents an address space in which a set of hosts is contained. A network object may derive its address and/or its mask from its member interfaces.

[0042] The Network Topology Discovery System With this in mind, **FIG. 3** refers to a network topology discovery system **300** according to an embodiment of the present invention. The system **300** is configured to be connected to an access point of a computer network, such as to stub network **112**, to send communications to and receive communications from hosts, typically routers. The system **300** comprises a phase controller **304** configured to oversee execution of each of the three phases in the appropriate order and determine when the discovery process is completed, an MIB discovery agent **308** configured in the MIB discovery phase to access the MIB information in each contactable router and generate an MIB output describing a portion of the network topology, and an OSP discovery agent **310** comprising an OSPF data collection agent **312** configured in the OSPF discovery phase to gather selected information regarding the network topology by contacting selected routers in each desired routing region and an OSPF data analyzing agent **316** configured in the OSPF discovery phase to analyze the information gathered by the OSPF data collection agent **312** and generate an OSPF output describing a portion of the network topology.

[0043] During topology discovery, the system **300** maintains a number of listings (outstanding, finished, and initial gateway lists, an area border router table (not shown), and a link state advertisement table (not shown)) to avoid duplication of computational effort. The outstanding list **320** (**FIG. 12**) lists candidate host (interface) addresses **1200** yet to be contacted. During the MIB and OSPF discovery phases, as new router interface addresses are discovered they are appended to the outstanding address list **320**. The MIB discovery phase will normally take the first address off the outstanding list and explore the router that the interface address is connected to. The finished list **324** (**FIG. 13**) lists host (interface) addresses **1300** that have been contacted and do not need to be contacted again. After a router interface on the outstanding address list has been processed, the corresponding router interface address is moved from the outstanding list to the finished list. The initial gateway list **328** (**FIG. 11**) lists OSPF routers **1100** (by a pointer to the pertinent router entry in the router table discussed below) and corresponding interface(s) **1104** (by a pointer to the pertinent interface entries in the interface table discussed below) discovered during operation of the MIB discovery agent **308**. These routers are explored during the OSPF discovery phase. The area border router table lists OSPF area border routers discovered during operation of the OSPF discovery agent **310** and includes, for each router entry, an indication of whether or not the routers have been contacted and, if so, the result. Finally, the link state advertisement table is output by the OSPF data collection agent **312** and is a listing of link state advertisements or LSAs obtained from the link state databases in the contacted OSPF routers. As will be appreciated, the link state database, as defined by the OSPF protocol, is a listing of links with each link being defined by end points and a cost metric associated with the link. Each area border router within a routing region has a complete copy of the database for all regions on whose border the router is located (or with which the area border router is associated). However, the non-area border routers within one routing region typically have a complete copy of the database in the region in which it is located and do not have the same link state database as a router in a different routing region.

[0044] The system **300** provides a number of output tables, namely the router, network interface, and interface tables **332**, **336**, and **340**, respectively, output by the MIB discovery agent **308** after completing the MIB discovery phase and the router, link, network, and interface lists **344**, **348**, **352**, and **356**, respectively, out put by the OSPF discovery agent **310** after completing the OSPF discovery phase. Referring to **FIG. 8**, the router table **332** contains, for each discovered router **800**, a corresponding protocol identifier **804** and one or more corresponding attached interfaces **808** (which are typically identified by a pointer to the corresponding entry in the interface table **340**). Referring to **FIG. 9**, the interface table **340** contains, for each interface **900**, a corresponding IP address **904**, one or more corresponding attributes **908** (e.g., whether the interface was contactable, speed, interface type, cost metrics, up/down status, etc.), and the corresponding router **912** owning the interface (which is typically identified by a pointer to the corresponding entry in the router table **332**). Referring to **FIG. 10**, the network interface table **336** lists, for each network address **1000** (or other type of network identifier), one or more corresponding interfaces **1004** (which are typically identified by a pointer to the corresponding entries in the interface table **340**) connected to the network. Referring to **FIG. 14**, the router list **344** contains a comprehensive listing of routers **1400**, both designated and attached, in the enterprise network or autonomous system. The routers are identified by router ID **1404** (and/or by one of the router's interface IP addresses (not shown)), associated area identifiers **1408**, and/or by one or more pointers **1412** referencing associated interfaces, in the interface list. As will be appreciated, an area border router will have multiple region IDs while a non-area border router will have only one, and a router can have one or more associated interfaces. Referring to **FIG. 15**, the link list **348** lists the discovered links. The links **1500** can have as endpoints **1504** two routers or a router and a network (stub or transit). The routers can be identified by router ID and/or interface IP address, and the network by mask and/or one or more IP addresses. Referring to **FIG. 16**, the network list **352** is a listing of networks **1600** (stub or transit). The networks **1600** are identified by a network address and mask and/or one or more IP addresses that are connected to the network. Each network has an associated set of router interfaces **1604** (which is typically indicated by a pointer to the corresponding router interface in the interface list), and an associated designated router **1608** (which is typically indicated by a pointer to the corresponding router in the router list). Finally referring to **FIG. 17**, the interface list **356** lists interfaces **1700** identified by IP address, interface number, and/or network mask. Each interface **1700** is associated with a router **1704**. The associated router **1704** is typically indicated by a pointer to the corresponding router in the router list **344**.

[0045] These tables collectively provide the network routing topology and the attributes of the network elements represented therein. As will be appreciated, "routing topology" refers to the logical network topology described by a particular routing protocol. Based on the router, interface and network interface tables and the router, link, network, and interface lists, a map or model of the routing topology can be generated automatically or manually. If more than one routing protocol is in use, there may be more than one distinct routing topology. As will be appreciated, the routing topology can be quite different from the physical network topology.

Operation of the Phase Controller

[0046] The operation of the phase controller **304** will now be described with reference to **FIG. 4**.

[0047] After creation of the phase controller **304** in step **400**, the phase controller **304** performs initial gateway detection in step **404**. The phase controller typically uses one or more seed IP addresses to contact a host router. The router(s) contacted initially by the phase controller **304** are typically referred to as a gateway router(s). In a preferred implementation, only one seed IP address is employed. If the user has not configured the phase controller to use a particular router as the initial gateway, the seed address can be determined automatically. The method by which this is determined is platform-dependent. For all platforms, the gateway is taken from the first routing table entry that has a valid gateway field. The Simple Network Management Protocol or SNMP techniques used to contact the gateway router can be routing protocol specific. For example, RFC1850 provides the specifications for contacting a router using the OSPF protocol. the phase controller **304** attempts to contact the initial gateway router in step **408**.

[0048] In decision diamond **412**, the phase controller **304** determines whether the gateway router was successfully contacted using SNMP techniques. When the contact attempt was not successful, the phase controller in step **416** terminates operation and notifies the user of an error and requests a further seed address. When the contact attempt is successful, the phase controller in step **420** accesses the MIB information in the initial gateway router and in step **424** initializes the outstanding and finished lists and appends the interface address of the contactable interface of the gateway router in the outstanding list for later use in the MIB discovery phase.

[0049] The phase controller in step **428** then calls the MIB discovery agent **308** to cause the agent to perform the MIB discovery phase. The operation of the MIB discovery agent is discussed below with reference to **FIGS. 5 and 6**.

[0050] When the MIB discovery phase is completed, the phase controller in step **432** calls the OSPF discovery agent **310** (or directly calls the OSPF data collection agent **312** first and then the OSPF data analyzing agent **316**) to perform the OSPF discovery phase. The operations of these agents are discussed below with reference to **FIG. 7**. The phase controller can use an observer pattern to monitor the progress of the MIB discovery agent and the OSPF discovery agent (or directly calls the OSPF data collection agent and the OSPF data analyzing agent).

[0051] When the OSPF discovery phase is complete, the phase controller determines in decision diamond **436** whether the outstanding list **320** is empty. If candidate entries **1200** remain in the outstanding list **320**, the phase controller repeats steps **428** and **432**. If no candidate entries **1200** remain in the outstanding list **320**, the phase controller proceeds to step **440** and terminates operation.

Operation of the MIB Discovery Agent

[0052] The operation of the MIB discovery agent **308** will now be discussed with reference to **FIGS. 5 and 6**. Generally, the MIB discovery agent **308** iteratively takes the next interface address from the outstanding list **320**, contacts the router corresponding to the interface address, processes the router's routing tables, and then moves the interface address from the outstanding list **320** to the finished list **324**. To avoid unnecessary network traffic and waste of computational resources, the MIB discovery agent is preferably configured so that when one interface of a router is contacted successfully the other interfaces of that router are not later contacted by the MIB discovery agent.

[0053] The MIB discovery agent **308** is created in step **500**. In decision diamond **504**, the MIB discovery agent **308** determines whether the outstanding list **320** is empty. When the list **320** is empty, the agent **308** ceases operation and the phase controller creates the OSPF data collection agent **312**. When the list **320** is not empty, the agent **308** proceeds to step **508** and gets the next interface address on the outstanding list **320**.

[0054] In step **512**, the agent **308** contacts the interface address, and in decision diamond **516** the agent **308** determines whether the address was contactable. When the address is not contactable, the agent proceeds to step **520** in which the interface address is moved from the outstanding list **320** to the finished list **324** and returns to decision **504** for the next iteration through the loop. When the address is successfully contacted, the agent **308** in step **524** processes the router as further described with reference to **FIG. 6**.

[0055] Referring to **FIG. 6**, processing of the router by the MIB discovery agent will now be described.

[0056] In step **600**, the MIB discovery agent **308** sets a flag associated with the interface address indicating that the address was successfully contacted. The flag may be contained in the outstanding list.

[0057] In step **604**, the agent **308** retrieves the list of interface addresses attached to the router corresponding to the contacted interface address. This information is contained in the router's SNMP tables. The SNMP tables retrieved by the agent are the ipAddr table, ipAddrTable table, ipRouteTable table, and ipRouteEntry table ("the Tables"), as defined in the IETF's RFC 1213 standard. As will be appreciated, the ipAddr table lists the IP addresses of the router's interfaces. The ipRouteTable table can be used to determine how each interface on the current router is connected to an interface on another router in the network (also referred to as the next hop interface). The table also describes the routing protocol, such as OSPF, present on the link between two router interfaces.

[0058] In step **608**, the agent retrieves the next interface address corresponding to the router (being processed) for processing. Prior to processing the address, the agent determines in decision diamond **612** whether the retrieved address is in the outstanding list **320**. When the retrieved address is in the outstanding list **320**, the address in step **616** is moved from the outstanding list **320** to the finished list **324**. Thereafter or when the retrieved address is not in the outstanding list **320**, the agent **308** then proceeds to step **620**.

[0059] In step **620**, the agent **308** retrieves the address of the next hop interface connected to the current interface of the router being processed. Based upon the information associated with the next hop interface, one or more entries is added to the network interface table **336**, the interface table **340**, and/or the router table **332**. If not done previously, the tables are initialized prior to addition of the entries.

[0060] The techniques used in step **624** to process the next hop interface addresses in the Tables will be known to those skilled in the art familiar with SNMP. As will be appreciated, the ipAddrTable::ipAdEntIfIndex acts as a key to each interface address specification in the ipAddrTable table. As shown in **FIG. 24**, the ipRouteEntry table is joined to the ipAddrTable table using the ipRouteEntry::ipRouteIfIndex as a foreign key (on ipRouteEntry::Instance). This relationship represents a one-to-many relationship between the ipAddrTable and ipRouteEntry table. For each ifAddrEntry there is 0 or more ipRouteEntry's that describe which network an interface is connected to and which interface is used to reach that network (using ipRouteEntry::ipRouteNextHop). If there are no ipRouteEntry's then the interface is not in use.

[0061] The tables can be used to differentiate a transport network from a stub network. When the packet can pass directly from the contacted interface to the network (or the interface is connected directly to a network (which is confirmed by the ipRouteType field designation of "direct")) the network is assumed to be a stub network. When a packet must pass from the contacted interface through a network to reach another network (or the interface is connected indirectly to the endpoint network listed in the ipAddrTable table (which is confirmed by the ipRouteType field designation of "indirect")), the intermediate network is a transport network.

[0062] Unnumbered interfaces must also be considered. For an interface to be unnumbered it must meet the criteria of appearing in the ipTable but not in the ipAddressTable, having an "Up" status in the ipTable, and not being an ethernet interface. Unnumbered interfaces will always be connected to another router using a point-to-link and to another unnumbered interface on the next router. In this case, the ipRouteTable on the current router will have a next hop address for the unnumbered interface as an arbitrary (numbered) interface address on the next router.

[0063] FIGS. **18-21** illustrate the ipAddrTable table, ipRouteEntry, and ipRouteTable tables. **FIG. 18** is the ipAddrTable table and contains, for each instance **1800**, ipAdEntAddr(IDX) **1804**, ipAdEntIfIndex **1806** and ipAdEntNetMask **1808**. As will be appreciated, an "instance" refers to an occurrence of an interface IP address. **FIG. 19** is the ipRouteEntry table and contains, for each instance **1900**, ipRouteDest(IDX) **1904**, ipRouteIfIndex **1908**, ipRouteNextHop **1912**, ipRouteType **1916**, ipRouteProto **1920** or the routing protocol supported by the connection between the subject interface and the next hop address **1912**, and ipRouteMask **1924**. **FIG. 20** is the ipAddrTable table and contains, for each instance **2000**, the same variables as the ipAddrTable table of **FIG. 18**. Finally, **FIG. 21** is the ipRouteTable table and contains, for each instance **2100**, the same variables as the ipRouteEntry table.

[0064] To illustrate the use of the above tables, several examples will now be discussed based on the tables of FIGS. 18-21. The interface address 192.168.34.1 or instance **1812** in the ifAddrEntry table **(FIG. 18)**, has an index of 4 and three joined entries **1902, 1928** and **1929** in the ipRouteEntry table **(FIG. 19)**. Looking at the entry **1928** shows that the network 192.168.34.0 can be reached from the 192.168.34.1 interface. This is logical as the interface address is a Class

C address ending in 34.1 and is confirmed by the ipRouteType field value, "Direct(3)". This interface is therefore directly connected to the 192.168.34.0 network and without further information the network is assumed to be a stub-network. The third entry **1929** shows that the network 192.168.35.0 can be reached via the interface 192.168.34.2 but, as shown by entry **1916**, is only indirectly connected as specified by the ipRouteType field "indirect(4)". The router that is the first hop towards this network can be reached by the next hop interface 192.168.34.2 as specified by the ipRouteNextHop field. To reach the 192.18.35.0 network a packet must pass through the 192.168.34.0 network, the 192.168.34.0 network is defined as a transport network. Since the 34.0 network was previously assumed to be a stub-network it now needs to be updated to be a transport network. On inspection of the ifAddrEntry (see **FIG. 20**) and ifRouteEntry (see **FIG. 21**) tables on the router with interface 192.168.34.2, it can be seen that the router connects to the 192.168.35.0 network via the 192.168.35.1 interface (see last entry in **FIG. 21**). This is determined by the 192.168.35.1 interface (entry **2002** in **FIG. 20**) in ifAddrEntry and the corresponding row in ifRouteEntry (ipRouteIndex =7) (entry **1908** in **FIG. 21**). The ipRouteType field **1916** specifies that this connection is "direct(3)" indicating that this interface is directly connected to the 35.0 network. The 192.168.34.2 router table (**FIG. 21**) also shows that this router is directly connected to the 17.0 network (see third entry from the top in **FIG. 21**) via the interface 192.168.17.2.

[0065] Returning to **FIG. 18**, there are three other interfaces that can also be explored. These are 192.168.16.100 (the original gateway interface), 192.168.19.2 and 192.168.29.2 interfaces. The joined ipRouteEntry table entries in **FIG. 19** for these interfaces show that this router is also connected to: (i) the 192.168.16.0 network directly via the 192.168.16.100 interface; (ii) the 192.168.19.0 network directly via the 192.168.19.2. interface; (iii) the 192.168.18.0 network indirectly via the 192.168.19.2 interface (and the next hop interface is 192.168.19.1 on a newly discovered router); and (iv) the 192.168.29.0 network directly via the 192.168.29.2 interface.

[0066] The network topology based on the foregoing analysis of FIGS. **18-21** is shown in **FIG. 22**

[0067] Returning again to **FIG. 6**, the agent **308** in step **628** determines whether the connection between the present interface and the next hop address uses OSPF. This is determined based upon the ipRouteProto **1920** entry in FIGS. **19** and/or **21**. When the connection supports OSPF, an OSPF protocol identifier is added to the router table **332** in step **629** and the interface address is added to the initial gateway list **328** in step **630** for later exploration in the OSPF discovery phase. In either event, the agent **308** next proceeds to decision diamond **632**.

[0068] In decision diamond **632**, the agent **308** determines whether the next hop address is in the outstanding or finished lists **320, 324**. When the next hop address is not in either list, the next hop address is added to the outstanding list **320** in step **636**. In either event, the agent **308** proceeds to decision diamond **640**.

[0069] In decision diamond **640**, the agent **308** determines whether there is another interface address attached to the contacted router that has not yet been considered. If another address has not yet been considered, the agent **308** returns to step **608** and repeats the above steps for that address. If no addresses remain to be considered, the agent **308** proceeds to step **520** of **FIG. 5**.

Operation of the OSPF Data Collection and Data Analyzing Agents

[0070] The operation of the OSPF data collection and data discovery agents **312** and **316** will now be discussed with reference to **FIG. 7**. Generally, each of the interface addresses in the initial gateway list **328** is used as an initial gateway starting point for the OSPF data collection agent **308** in the OSPF discovery phase. OSPF routers that were discovered by a previous OSPF discovery run do not need to be used as initial gateway starting points on subsequent runs of the OSPF data collection agent **312**.

[0071] Referring to **FIG. 7**, the phase controller **304** in step **700** retrieves the initial gateway list **328** and gets the next router (or the first interface address attached to the router) on the list **328** in step **704** to process in OSPF discovery. The phase controller then sets the OSPF discovery initial gateway to the first interface address in step **708** and creates the OSPF data collection agent **312** in step **712** and then the OSPF data analyzing agent **316** in step **716**. The operation of the data collection agent **312** and data analyzing agent **316** are described in U.S. patent application, Ser. No. ______, filed concurrently herewith, entitled "USING LINK STATE INFORMATION TO DISCOVER IP NETWORK TOPOLOGY", to Goringe, et al., and U.S. Provisional Application Serial No. 60/317,719, filed Sep. 6, 2001, of the same title, to Goringe, et al., each of which is incorporated herein by this reference.

[0072] As set forth in the above applications, the data collection agent **312** uses a seed interface IP address to contact a host router in a selected routing area, downloads the ospfLsdb Table in the link state database of the contacted host router, discards any link state advertisements outside the area(s) of interest, adds the IP address of each interface associated with an area border router to the area border router table, and adds any LSA's for area(s) of interest to the link state advertisement table. These steps are repeated for each area border router.

[0073] The data analyzing agent **316** traverses the link state advertisement table discarding all link state advertisements other than types 1 and 2 and, using the type 1 and 2 link state advertisements, forms the router list **344 (FIG. 14)** (which contains all discovered OSPF routers in the areas of interest), the link list **348 (FIG. 15)**, the network list **352 (FIG. 16)**, and/or the interface list **356 (FIG. 17)**.

[0074] After the operations of the data collection and data analyzing agents **312** and **316** are completed, the phase controller **304** in step **720** removes all discovered OSPF routers from the initial gateway list **328**. As will be appreciated, the data collection agent **312** can set a flag for each discovered OSPF router.

[0075] In step **724**, the phase controller retrieves all router interface addresses found by the data collection and analyzing agents **312** and **316** to ascertain whether further MIB discovery phase operation is required. This determination is made to ensure consideration by the MIB discovery agent **308** of any router interface addresses that were discovered during OSPF discovery and not MIB discovery. To this end, the phase controller **304** in step **728** gets the next retrieved interface address and in step **732** determines whether the retrieved interface address is in the outstanding or finished lists **320** and **324**. When the address is not in the outstanding or finished lists **320** and **324**, the controller adds the address to the outstanding list in step **736**. In decision diamond **740**, the controller **304** then determines whether there are additional interface addresses for consideration. If so, the controller returns to step **728** and repeats the foregoing steps. If not, the controller proceeds to decision diamond **744** and determines whether there are additional OSPF routers in the initial gateway list **328**. When additional OSPF routers are in the list **328**, the controller returns to step **704** for further OSPF discovery phase operation. When additional routers are not in the list **328**, the controller proceeds to step **436** of **FIG. 4**.

[0076] The various output tables and lists can be merged in several ways to form a consolidated network topology model. In an "as you go" approach, the network model is a knowledge repository. The discovery algorithms deal with the model by propositional methods, namely the algorithms inform the model what they have found out and leave it to the model to decide what to do with it. When information is received from a protocol which indicates that previously obtained information about a network or a router and the like is incomplete, the additional information is added to the model. All information is marked with which protocols provided it to assist in the identification of conflicting information. Where information from two different protocols is contradictory, the topology information from MIB2 routing tables is preferred. The information is marked as contradicted by the pertinent routing protocol and the routing protocol information is retained.

[0077] A number of variations and modifications of the invention can be used. It would be possible to provide for some features of the invention without providing others.

[0078] For example in one alternative embodiment, the algorithm is used for a protocol other than OSPF, for non-SNMP network management protocols such as the Common Management Information Protocol or CMIP and/or for discovery techniques not relying on the access of MIB information. The algorithm can be used to discover any router supporting any distance-vector and link-state routing protocol.

[0079] In another alternative embodiment, the algorithm is used simultaneously for multiple routing protocols. **FIG. 243** illustrates this embodiment. In step **2400** (when the gateway router is contacted), the system **300** detects the routing protocol in use on the target network. In some configurations, the gateway router can be using more than one routing protocol. In steps **2404, 2408** and **2412** topology discovery is performed using a different algorithm for each detected routing protocol. In **FIG. 243, a** different algorithm is used for each of the OSPF, RIP, and EIGRP routing protocols. The algorithms for the RIP and EIGRP will be a modified form of the algorithms discussed above, with the modifications being obvious to one of ordinary skill in the art. If more than one routing protocol is in use, there may be more than one routing topology. The network model of the present invention may not merge the differing routing topologies into a common model. In that event, each routing topology is concurrently represented in a single data structure.

[0080] In another alternative embodiment, the various discovery agents are called on a router-by-router basis depending upon what routing protocol the router supports. For example, during topology discovery an OSPF can discover a first OSPF router and an RIP agent can then be called to discover the next router which supports RIP but not OSPF.

[0081] In another alternative embodiment, the MIB and OSPF discovery phases can be performed in reverse order, namely OSPF discovery can be performed before MIB discovery.

[0082] In another alternative embodiment, the MIB and OSPF discovery phases are conducted in parallel or simultaneously.

[0083] In another embodiment, any of the software modules discussed above can be implemented, in whole or part, as an application specific integrated circuit or any other type of logic circuit.

[0084] The present invention, in various embodiments, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various embodiments, subcombinations, and subsets thereof. Those of skill in the art will understand how to make and use the present invention after understanding the present disclosure. The present invention, in various embodiments, includes providing devices and processes in the absence of items not depicted and/or described herein or in various embodiments hereof, including in the absence of such items as may have been used in previous devices or processes, e.g. for improving performance, achieving ease an\or reducing cost of implementation.

[0085] The foregoing discussion of the invention has been presented for purposes of illustration and description. The foregoing is not intended to limit the invention to the form or forms disclosed herein. Although the description of the invention has included description of one or more embodiments and certain variations and modifications, other variations and modifications are within the scope of the invention, e.g. as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative embodiments to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method for discovering a topology of a distributed processing network, comprising:

(a) retrieving from a first set of routers a first type of information stored in each router in the first set of routers; and

(b) retrieving from a second set of routers a second type of information stored in each router in the second set of routers, wherein the first and second sets of routers are different and the first and second types of information are different.

2. The method of claim 1, wherein the first set of routers includes at least some of the routers in the second set of routers.

3. The method of claim 1, wherein the first type of information comprises one or more variables in a management information base and the second type of information comprises one or more variables defined by a routing protocol.

4. The method of claim 1, further comprising:

maintaining an outstanding list of uncontacted routers and/or router interfaces during the contacting steps (a) and (b).

5. The method of claim 4, further comprising:

maintaining a finished list of contacted routers and/or router interfaces during the contacting steps (a) and (b).

6. The method of claim 1, further comprising:

retrieving a next hop interface address associated with a selected router and/or interface;

determining a routing protocol supported by the next hop address; and

assigning to at least one of the selected router and/or interface and the next hop interface address a protocol identifier indicating the routing protocol.

7. The method of claim 1, further comprising:

maintaining an initial gateway list during contacting step (a) for use in contacting step (b).

8. The method of claim 1, further comprising:

based on at least one of the first and second types of information, identifying one or more of interface objects, router objects, and network objects associated with the distributed processing network

9. The method of claim 1, wherein in retrieving step (a) the first type of information is defined by a network management protocol and in retrieving step (b) the second type of information is defined by a routing protocol.

10. A method for discovering a topology of a distributed processing network, comprising:

(a) retrieving from a first set of routers a first type of information stored in each router in the first set of routers, the first type of information being defined by a network management protocol; and

(b) retrieving from a second set of routers a second type of information stored in each router in the second set of routers, the second type of information being defined by a routing protocol.

11. The method of claim 10, wherein the first and second sets of routers are different and the first and second types of information are different.

12. The method of claim 11, wherein the first set of routers includes at least some of the routers in the second set of routers.

13. The method of claim 10, wherein the first type of information comprises one or more variables in a management information base defined by the simple network management protocol and the second type of information comprises one or more variables defined by a routing protocol.

14. The method of claim 10, further comprising:

maintaining an outstanding list of uncontacted routers and/or router interfaces during the contacting steps (a) and (b).

15. The method of claim 14, further comprising:

maintaining a finished list of contacted routers and/or router interfaces during the contacting steps (a) and (b).

16. The method of claim 10, further comprising:

retrieving a next hop interface address associated with a selected router and/or interface;

determining a routing protocol supported by the next hop address; and

assigning to at least one of the selected router and/or interface and the next hop interface address a protocol identifier indicating the routing protocol.

17. The method of claim 10, further comprising:

maintaining an initial gateway list during contacting step (a) for use in contacting step (b).

18. The method of claim 10, further comprising:

based on at least one of the first and second types of information, identifying one or more of interface objects, router objects, and network objects associated with the distributed processing network.

19. The method of claim 1, further comprising:

(c) processing at least one of the routers in the first set of routers by performing at least the following steps:

retrieving a list of interface addresses stored in the processed routers and for each interface address in the list of interface addresses, iteratively performing steps of:

(i) selecting an address of a next hop interface connected to the selected interface address;

(ii) identifying at least one of a router object, an interface object, a network object, and a link object associated with the selected interface address;

(iii) identifying a protocol associated with the selected next hop interface; and

(iv) repeating steps (i)-(iii) for each next hop interface associated with the selected interface address.

20. A system for discovering a topology of a distributed processing network, comprising:

(a) first means for contacting a first set of routers to obtain a first type of information stored in each router in the first set of routers; and

(b) second means for contacting a second set of routers to obtain a second type of information stored in each router in the second set of routers, wherein the first and second sets of routers are different and the first and second types of information are different.

21. The system of claim 20, wherein the first set of routers includes at least some of the routers in the second set of routers.

22. The system of claim 20, wherein the first type of information comprises one or more variables in a management information base and the second type of information comprises one or more variables defined by a routing protocol.

23. The system of claim 20, further comprising:

means for maintaining an outstanding list of uncontacted routers and/or router interfaces for use by the first contacting means.

24. The system of claim 23, further comprising:

means for maintaining a finished list of contacted routers and/or router interfaces.

25. The system of claim 20, wherein the first contacting means comprises:

retrieving means for retrieving a next hop interface address associated with a selected router and/or interface;

determining means for determining a routing protocol supported by the next hop address; and

assigning means for assigning to at least one of the selected router and/or interface and the next hop interface address a protocol identifier indicating the routing protocol.

26. The system of claim 20, further comprising:

means for maintaining an initial gateway list for use by the second contacting means.

27. The system of claim 20, further comprising:

based on at least one of the first and second types of information, means for identifying one or more of interface objects, router objects, and network objects associated with the distributed processing network

28. The system of claim 20, wherein the first type of information is defined by a network management protocol and the second type of information is defined by a routing protocol.

29. A system for discovering a topology of a distributed processing network, comprising:

(a) a first topology discovery agent configured to contact a first set of routers to obtain a first type of information stored in each router in the first set of routers;

(b) a second topology discovery agent configured to contact a second set of routers to obtain a second type of information stored in each router in the second set of routers, wherein the first and second sets of routers are different and the first and second types of information are different; and

(c) a phase controller configured to select between the first and second topology discovery agents.

30. The system of claim 29, wherein the first set of routers includes at least some of the routers in the second set of routers.

31. The system of claim 29, wherein the first type of information comprises one or more variables in a management information base and the second type of information comprises one or more variables defined by a routing protocol.

32. The system of claim 29, wherein at least one of the first and second topology discovery agents and the phase controller are configured to maintain an outstanding list of uncontacted routers and/or router interfaces for use by the first topology discovery agent.

33. The system of claim 29, wherein at least one of the phase controller and first and second topology discovery agents is configured to maintain a finished list of contacted routers and/or router interfaces.

34. The system of claim 29, wherein the first topology discovery agent is configured to retrieve a next hop interface address associated with a selected router and/or interface, determine a routing protocol supported by the next hop address, and assign to at least one of the selected router and/or interface and the next hop interface address a protocol identifier indicating the routing protocol.

35. The system of claim 29, wherein at least one of the phase controller and first and second topology discovery agents is configured to maintain an initial gateway list for use by the second contacting means.

36. The system of claim 29, wherein at least one of the phase controller and first and second topology discovery agents is configured to identify one or more of interface objects, router objects, and network objects associated with the distributed processing network.

37. A network topology model, comprising:

a plurality of interface identifiers, each interface identifier being associated with a corresponding router;

a plurality of network identifiers; and

a plurality of routing protocol identifiers, each routing protocol identifier being associated with at least one of a router and an interface and being indicative of a routing protocol supported by the at least one of a router and an interface.

38. The network topology model of claim 37, wherein each interface identifier is an electronic address on a network.

39. The network topology model of claim 37, wherein each network identifier is a network address.

40. The network topology model of claim 37, wherein the protocols associated with the routing protocol identifiers comprise at least two distance-vector algorithms, link-state algorithms, and combinations thereof.

41. A method for determining a topology associated with an enterprise network, comprising:

identifying at least one routing protocol used in the enterprise network; and

based on the identified routing protocol, selecting a data collection agent from among a plurality of data collection agents to collect network topology information, wherein each of the plurality of data collection agents is configured to support a different routing protocol.

42. The method of claim 41, wherein the enterprise network uses first and second routing protocols and the first routing protocol is associated with a first data collection agent and the second routing protocol is associated with a second, different routing protocol and further comprising:

collecting a first set of network information using the first data collection agent and a second, different set of network information using the second data collection agent.

43. The method of claim 42, further comprising:

forming the first set of network information into a first network model and the second set of network information into a second, different network model.

44. The method of claim 41, further comprising:

merging the first and second network models into a single network model.

* * * * *

CREDENTIAL MANAGEMENT AND NETWORK QUERYING

CROSS REFERENCE TO RELATED APPLICATIONS

The present application claims priority under 35 U.S.C.§119 to U.S. Provisional Application Serial No. 60/347,060, of the same title and filed January 8, 2002, to Goringe, et al., which is incorporated herein by this reference.

FIELD OF THE INVENTION

The present invention is related generally to authentication in data networks and specifically to determining credentials for computational components in data networks.

BACKGROUND OF THE INVENTION

In computational networks, it is common to have one or more automated network management system (NMS) devices for collecting data to ascertain levels of performance (*e.g.*, BER, loss of synchronization, *etc.*), equipment, module, subassembly, and card failures, circuit outages, levels of traffic, and network usage. NMS devices typically interrogate network components, such as routers, ethernet switches, and other hosts for stored information. As will be appreciated, a network device or component is a computational component that may or may not have a physical counterpart, *e.g.*, the component may be a virtual computational component such as an interface. Examples of proprietary network management systems include Hewlett-Packard's OPENVIEW™, IBM's NETVIEW™, and Digital Equipment Corporation's EMA™. To permit such network management systems in distributed processing networks to communicate with hosts for monitoring and controlling the enterprise network, network management communication protocols have been developed, such as the Simple Network Management Protocol or SNMP and the Common Management Information Protocol or CMIP.

During interrogation, NMS devices interact with authentication systems present in network devices, such as routers. Authentication systems are an essential part of network security. Typically, a user is able to access information in certain network devices only by entering one or more credentials. As used herein, a "credential" refers to a set of information (e.g., a character or string of characters) which must be provided to a computational component for access to information in the computational component to be provided. Examples of credentials for version 1 of SNMP include a community string, for version 3 of SNMP User-Based/Security Model or include USM mode, user name, authentication method, authentication password, privacy method, and privacy password, and for TELNET include a user login, password, router type, and prompt. As will be appreciated, different credentials can be required for differing levels of information access, *e.g.* read-only access and supervisor levels.

When a new NMS system device is connected to a network, the NMS device must learn the various forms of authentication used to be able to interrogate network devices. The learning process typically involves a user manually setting credentials before using the tool on the network. This is not only a slow task but also fails to easily allow for dynamic changes of authentication during use. For example, some network security schemes require a credential to be periodically changed to maintain a high level of network security.

Network management personnel typically compromise network security for ease of credential configuration in NMS devices. For example, some network management systems rely on the credential being set to a default credential (generally public level access credentials) on all components in the network. In some applications, the varying access levels to the network components are compromised by using a common default credential. This practice unnecessarily restricts the type of authentication to a type of default credential and can restrict with what type of equipment the network management system can be used and also compromises network security. Other network management systems do permit a limited number of passwords to be entered before the network management system performs interrogation but fail to allow for dynamic changes in authentication during use.

SUMMARY OF THE INVENTION

These and other needs are addressed by the various embodiments and configurations of the present invention. The credential discovery agent of the present invention determines credentials of network devices by maintaining a credential repository, which typically is a historical record of credentials used in the network, and/or a candidate credential queue, which typically is a listing of credentials ordered based on the likelihood that the credentials are in current use by the network devices of interest. In one architecture, the agent, repository, and queue consider that network management personnel reuse credentials over time and, at any given time, reuse the same credential for different network devices.

In one embodiment, the credential discovery agent determines one or more credentials of a network device by performing the steps of:

(a) selecting a first network device from among a plurality of network devices;

(b) accessing the credential repository, the credential repository comprising a first set of credentials corresponding to the first network device;

(c) contacting the first network device; and

(d) testing the validity of the first set of credentials.

The credential repository holds credentials that have been learned (e.g., from the user, by a successful guess, etc.). The repository is used to save the credentials between executions and can have things removed or added to it during agent operation. Between runs the repository allows the credentials to be stored so they can be used on subsequent runs of the agent.

The credential repository can include a number of variables associated with the first network device. These variables can include a corresponding credential state, a corresponding protocol identifier, a corresponding (IP) address, a total number of instances of use of at least one credential in the first set of credentials, a corresponding candidate credential frequency counter associated with at least one credential in the first set of credentials, a recency of use of at least one credential in the first set of credentials, and the administrative locality of at least one credential in the first set of credentials. The protocol identifier is indicative of the protocol defining or associated with the credentials and/or the authentication system used to communicate with the network device.

If the agent is unable to determine the valid credentials using the repository, the agent can prompt the user for additional credentials to test. In this manner, the user can provide input into the operation of the agent. The user is typically prompted for credentials as the agent contacts differing types of network devices. The user fills in the required credential(s) and the agent then verifies that the inputted credential(s) are correct by using the inputted credential(s) to contact the network device. When the credential(s) is valid, it is copied into the repository.

In another embodiment, the agent determines at least one credential of a network device when previously used credentials are invalid or unsuccessfully validated by performing the steps of:

(a) selecting one or more credential from a candidate credential queue;

(b) contacting a network device;

(c) testing the validity of the credential(s); and

(d) assigning a priority value or ranking to the tested credential based on whether or not the credential(s) is valid.

The priority value is used to determine an order in which to test corresponding credentials when it is necessary to guess the credential in use by the network device. In one configuration, the priority value is used to order the listing of credentials in the candidate credential queue. In another configuration, the priority value is determined based on one or more of a candidate credential frequency counter, a recency of use counter, and an administrative locality associated with the corresponding set of credentials.

In one configuration, the agent attempts to guess the credential before prompting the user for a credential. These guesses may include standard defaults, credentials which have been used or tried elsewhere in the network, or credentials which have been provided by the user up-front.

The agent, credential repository, and candidate credential queue can have a number of advantages. First, the agent can dynamically and automatically maintain the repository and candidate over time. Conventional tools allow for a limited number of credentials to be entered before the tool is used, but such tools do not allow for dynamically adding more credentials during use of the tool. In contrast, the agent updates the repository and queue during and/or after each run of the credential discovery agent. Second, the agent can be convenient to use and determine credentials in significantly less time than conventional techniques. Third, the agent can reduce the amount of user interaction by making educated guesses at the credential before prompting the user. In some configurations, the agent speculatively tests credentials on any new network devices detected to reduce the requirement for user interaction. Fourth, the agent can obviate the need for the user to manually input an extensive list of credentials before the agent is run. Fifth, the agent can make network management systems more flexible in dealing with unknown credentials by prompting the user and also storing known credentials in the repository for later use. These and other advantages will be apparent from the disclosure of the invention(s) contained herein.

The above-described embodiments and configurations are neither complete nor exhaustive. As will be appreciated, other embodiments of the invention are possible utilizing, alone or in combination, one or more of the features set forth above or described in detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a computational architecture according to a first embodiment of the present invention and

Figs. 2A and 2B depict a flow schematic of the credential discovery agent.

DETAILED DESCRIPTION

Fig. 1 depicts a computational architecture 100 according to a first embodiment of the present invention. The architecture 100 comprises a credential discovery agent 104 configured to determine one or more valid credentials for selected network devices or components, a credential repository 108 mapping credentials to IP addresses and containing other information, a candidate credential queue 112 listing credentials in order of priority for credential guessing by the credential discovery agent 104, and a skip list 116 listing IP addresses for which credential determination was not performed at the request of the user.

The credential repository 108, which is typically encrypted, is loaded at runtime of the agent 104 to provide an initial population of credentials for IP addresses of network components. The repository can include a number of fields for each IP address including one or more credentials, a credential state, a protocol identifier, and protocol access level for credential and/or for each credential a protocol identifier, corresponding IP addresses, the total number of instances of use of the credential by the listed IP addresses, a priority of use of the credential, a candidate credential frequency counter to reflect the frequency of use of the credential in the network (or in the credential repository), recency of use of the (valid) credential in the network (or recency of use as determined by the agent 104), the administrative locality of the credential, and other information that can be used to assign a priority value to the credential in the candidate credential queue 112. During operation of the agent 104, the credential repository 108 is updated by the agent 104, such as after each IP address is considered and/or after all of the IP addresses are considered. As will be appreciated, a unique network component identifier other than IP address can be employed, depending upon the protocol associated with the network component.

The candidate credential queue 112 provides a listing of credentials, each of which has a corresponding priority and protocol identifier. When guessing, the agent 104 tests the credentials in order of each credential's corresponding priority value. In one implementation for version 1 of SNMP, the queue 112 is initially populated with a credential containing the community string "public". During any individual discovery task, each credential, which is successfully validated by the credential repository is also added to the queue 112, though with a lower priority than that of the "public" credential. As will be appreciated, the priority can be assigned based on any one of or combination of factors including the candidate credential frequency counter to reflect the frequency of use of the credential in the network (or in the credential repository), the recency of use of the (valid) credential in the network (or the recency of use as determined by the agent 104), and/or the administrative locality of the credential relative to the IP address under consideration (*e.g.,* if the network component under consideration is associated with or connected to another network component which has a corresponding credential the corresponding credential is first used as a test credential).

The skip list 116 is simply a listing of network component IP addresses for which the agent 104 will not perform a credential determination.

The operation of the credential discovery agent 104 is depicted in Figs. 2A and 2B. Referring to Fig. 2A, the agent 104 is created in step 200.

In step 204, the agent 104 determines if the credential repository 108 is populated with one or more IP addresses. If the credential repository 108 is empty or nonexistent, the agent 104 initializes the repository and proceeds to step 208. If the credential repository is not empty, the repository is loaded by the agent in step 212. Initially, all credentials in the credential repository 108 are assumed to be untested or not yet successfully validated. The agent 104 then proceeds to step 208.

In decision diamond 208, the agent determines whether the user has requested to stop discovery. If the user has so requested, the agent 104 proceeds to step 216 and returns with an error code (STOP_CRED) indicating the request. If the user has not so requested, the agent proceeds to step 220.

In step 220, the agent selects an initial IP address for credential determination. The initial IP address is typically selected from a network access list of one or more IP addresses provided by the user. This network access list can be generated by the user manually or automatically using a network topology discovery algorithm such as described in U.S. Patent Applications entitled "Topology Discovery by Partitioning Multiple Discovery Techniques" and "Using Link State Information to Discover IP Network Topology", both by Goringe, et al., filed concurrently herewith and incorporated herein by this reference. The network access list typically includes a list of network component identifiers (*e.g.*, IP addresses) and a corresponding credential state field for each identifier.

The agent then proceeds to step 224 where the agent determines if the selected IP address is on the skip list 116.

If the selected IP address is on the skip list 116, the agent 104 sets the credential state for the IP address in the network access list as NO CREDENTIAL in step 228 and proceeds to decision diamond 232 where the agent determines if there is another IP address on the network access list. The NO CREDENTIAL state means that no valid credential was obtained for the corresponding IP address. The corresponding IP address entry in the credential repository 108 (if any) is typically not removed from the repository if the IP address is skipped. If a next IP address is available, the agent 104 gets the next IP address in step 236 and repeats step 224. If a next IP address is unavailable, the agent 104 saves the updated credential repository and terminates operation in step 216.

If the IP address is not on the skip list, the agent 104 next determines in decision diamond 240 whether there is in the credential repository 108 an IP address entry matching the selected IP address. In other words, the agent 104 determines whether the repository 108 contains a credential corresponding to the selected IP address.

When a corresponding credential exists, the agent in step 244 tests the validity of the credential by known techniques. The techniques, of course, depend upon the protocol being used by the network component corresponding to the IP address.

When the credential is valid in step 248, the agent 104 proceeds to step 252 where the credential is added to the candidate credential queue 112 and then to step 256 where the corresponding entry in the network access list (and/or credential repository) is assigned the credential state of FOUND CREDENTIAL. This state means that the credential was validated. The credential is stored in the appropriate out-parameter corresponding to the IP address. The agent 104 may increment a candidate credential frequency counter and/or otherwise adjust the priority of the credential in the candidate credential queue 112. The agent 104 then returns to step 232 discussed above.

When the credential is invalid in step 248, the agent 104 must determine the reason why the credential was not successfully validated. The unsuccessful validation could be due to an invalid credential or to the network component being uncontactable at the time. Accordingly, the agent 104 in step 260 pings the device and in decision diamond 264 determines whether a response is received from the component within a selected time interval. The ping step 260 can be done using an Internet Control Message Protocol or ICMP echo request.

In any event, if a response is not received, the agent 104 in step 268 assigns a credential state of UNCONTACTABLE to the corresponding entry in the network access list (and/or credential repository) and returns to step 232 above. As will be appreciated, the credential state of UNCONTACTABLE indicates that the network component was unresponsive to the ping. The corresponding IP address entry in the credential repository is not removed when the credential state is UNCONTACTABLE.

If a response is received, the agent 104 in step 272 removes the entry corresponding to the IP address from the credential repository 108, updates the entry corresponding to the credential in the credential repository 108, and adjusts the candidate credential queue 112 when the credential is listed in the candidate credential queue. As noted, the priority of the credentials in the queue 112 can be based on any number of factors, including usage of the credential. When the credential is no longer in use by a network component, the priority often requires adjustment downward to reflect the nonuse. Typically, the candidate credential frequency counter is decremented.

The agent next proceeds to step 276 where the agent 104 attempts to guess the credential from the credentials listed in the queue 112. When guessing, the agent 104 tries all of the credentials in the queue 112 in order of priority. As shown in steps 280, 284, and 288, each credential is retrieved sequentially and an attempt is made to validate it.

When a credential is successfully validated in steps 276, 280, 284 and 288, the credential is stored in the appropriate out-parameter corresponding to the IP address in step 292 and the corresponding entry in the network access list (and/or credential repository) is assigned the credential state of FOUND CREDENTIAL in step 256. The agent 104 may increment a candidate credential frequency counter and/or otherwise adjust the priority of the credential in the candidate credential queue 112. The agent 104 then returns to step 232 which is discussed above.

When a credential is unsuccessfully validated in steps 276, 280, 284 and 288, the agent 104 in step 296 checks the user's preferences regarding whether or not the user is to be prompted for further instructions regarding the IP address. This preference is indicated by using a flag state. If no credentials that can be used to access the remote network component are found or if none of the found credentials work, the user may be prompted for a new set of credentials. The user is prompted only if the existence of the remote network component has earlier been confirmed by pinging as noted above and the flag to not prompt the user is not set (or vice versa).

In decision diamond 300, the agent 104 determines whether to prompt the user. When the prompt flag is set(*i.e.,* the user does not want to be prompted) then the agent 104 in step 304 marks the IP address for which no credential can be found as through the user had responded with a skip command. In other words, the IP address is added to the skip list 116. The corresponding entry in the network access list (and/or credential repository) is then assigned in step 308 a credential state of NO CREDENTIAL. The agent 104 then returns to step 232 discussed above.

When the prompt flag is not set(*i.e.,* the user wants to be prompted), then the agent 104 in step 312 prompts the user. The user can respond in five different ways. First, the user can respond by entering a credential as shown by decision diamond 316. When a credential is entered, the agent 104 tests the validity of the credential in step 320. When in step 324 the credential is valid, the agent proceeds to step 292 discussed above. When in step 324 the credential is invalid, the agent returns to step 312 and again prompts the user. Second, the user can respond by instructing the agent 104 to skip the IP address. This is shown in step 328. When the agent 104 receives this response, the agent 104 proceeds to step 304 discussed previously. Third, the user can respond by instructing the agent 104 to stop. This is shown in step 332. In that event, the agent 104 sets the prompt flag to stop in step 336, adds the address to the skip list 116 in step 340, saves the updated credential table and terminates operation in step 344. Fourth, the user can respond by instructing the agent 104 to no prompt. This is shown by step 348. In that event, the agent 104 sets the prompt flag to no prompt in step 352 and proceeds to step 304 discussed above. Finally, the user can provide an unintelligible or unrecognized response. In that event, the agent 104 returns to step 312 and again prompts the user.

Returning to decision diamond 240, when a corresponding credential is not in the credential repository the agent 104 in step 356 pings the device as discussed above to determine if the network component is contactable. The agent 104 in decision diamond 360 determines whether or not a response is timely received. When a timely response is received, the agent 104 proceeds to step 276 discussed above. When no timely response is received, the agent 104 proceeds to step 268 also discussed above.

A number of variations and modifications of the invention can be used. It would be possible to provide for some features of the invention without providing others. For example in one alternative embodiment, the architecture discussed above supports other versions of SNMP, such as version 3 of SNMP, and/or protocols other than SNMP, such as TELNET and CMIP. In this embodiment, the credential object would be defined in way(s) to support one or more different protocols. For example, the architecture can support multiple protocols at the same time. A protocol identifier is then used in the credential repository to identify the protocol corresponding to the network component and the credential object accorded a number of alternative definitions depending upon the corresponding protocol. In this embodiment, the credentials in the candidate credential frequency queue 112 would only be used in the credential guessing routine for the network component corresponding to the IP address under consideration when the network component used the protocol corresponding to the credential (as shown by the corresponding protocol identifier). In another alternative embodiment, a unique network component identifier other than IP address is used in the credential repository. For example, the identifier could be a component id as defined by the OSPF protocol, and/or credentials preconfigured by the user to be used as candidates for guessing. In another alternative embodiment, credentials in the repository that are not successfully validated are not removed from the respository but are marked with an appropriate flag indicating this fact. The credential may still be used by the network at a subsequent time or be concurrently used by a network component that is not listed in the credential repository. These credentials are eligible for inclusion in the candidate credential queue 112. As will be appreciated, some network security schemes rotate use of or periodically reuse credentials. In yet another alternative embodiment, the candidate credential queue can include credentials from sources other than the network itself. For example, the queue can include credentials that are in common or widespread use in the industry, default credentials in use when a device is initially acquired from a supplier or manufacturer, and/or credentials that are provided by the user in advance.

The present invention, in various embodiments, includes components, methods, processes, systems and/or apparatus substantially as depicted and described herein, including various embodiments, subcombinations, and subsets thereof. Those of skill in the art will understand how to make and use the present invention after understanding the present disclosure. The present invention, in various embodiments, includes providing devices and processes in the absence of items not depicted and/or described herein or in various embodiments hereof, including in the absence of such items as may have been used in previous devices or processes, e.g. for improving performance, achieving ease and\or reducing cost of implementation.

In one alternative embodiment, the credential discovery agent is implemented in whole or part as an application specific integrated circuit or other type of logic circuit.

The foregoing discussion of the invention has been presented for purposes of illustration and description. The foregoing is not intended to limit the invention to the form or forms disclosed herein. Although the description of the invention has included description of one or more embodiments and certain variations and modifications, other variations and modifications are within the scope of the invention, e.g. as may be within the skill and knowledge of those in the art, after understanding the present disclosure. It is intended to obtain rights which include alternative embodiments to the extent permitted, including alternate, interchangeable and/or equivalent structures, functions, ranges or steps to those claimed, whether or not such alternate, interchangeable and/or equivalent structures, functions, ranges or steps are disclosed herein, and without intending to publicly dedicate any patentable subject matter.

What is claimed is:

1. A method for determining one or more credentials of a network device, comprising:

selecting a first network device from among a plurality of network devices;

accessing a credential repository, the credential repository comprising a first set of credentials corresponding to the first network device;

contacting the first network device; and

testing the validity of the first set of credentials.

2. The method of Claim 1, wherein the credential repository further comprises, for the first network device, a plurality of a corresponding credential state, a corresponding protocol identifier, a corresponding address, a total number of instances of use of at least one credential in the first set of credentials, a corresponding candidate credential frequency counter associated with at least one credential in the first set of credentials, a recency of use of at least one credential in the first set of credentials, and the administrative locality of at least one credential in the first set of credentials.

3. The method of Claim 1, further comprising:

assigning a credential state to at least one credential in the first set of credentials.

4. The method of Claim 1, further comprising, when at least one credential in the first set of credentials is valid:

adding the at least one credential to a candidate credential queue.

5. The method of Claim 1, further comprising, when at least one credential in the first set of credentials is not valid:

pinging the first network device to determine whether the first network device is contactable.

6. The method of Claim 1, further comprising, when at least one credential in the first set of credentials is not valid:

selecting a second set of credentials from a candidate credential queue; and

testing the validity of the second set of credentials.

7. The method of Claim 6, further comprising, when at least one credential in the second set of credentials is not valid:

prompting a user for a third set of credentials; and

when the third set of credentials is received from the user, testing the validity of the third set of credentials.

8. The method of Claim 1, wherein the plurality of network devices comprises a second network device, each of the first and second network devices has an associated protocol identifier indicative of a protocol used by the network device, and the first and second protocol identifiers are different.

9. The method of Claim 6, further comprising:

comparing a protocol associated with the first network device with a protocol identifier associated with the second set of credentials.

10. The method of Claim 9, wherein, when the protocol associated with the first network device is different from the protocol associated with the protocol identifier, the testing step is not performed.

11. A credential repository for a network, comprising:

a plurality of network device identifiers, at least first and second network devices in the plurality of network devices using different protocols;

for each of the plurality of network devices, the credential repository further comprises, for the first and second network devices, a corresponding protocol identifier; and

for each of the plurality of network devices, a corresponding set of credentials configured for the protocol associated with the corresponding protocol identifier.

12. The credential repository of Claim 11, comprising, for each of the plurality of network device identifiers, a plurality of a corresponding credential state, a corresponding address, a total number of instances of use of at least one credential in the corresponding set of credentials, a corresponding candidate credential frequency counter associated with at least one credential in the corresponding set of credentials, a recency of use of at least one credential in the first set of credentials, and an administrative locality of at least one credential in the corresponding set of credentials.

13. A system for determining one or more credentials of a network device, comprising:

means for selecting a first network device from among a plurality of network devices;

a credential repository comprising a first set of credentials corresponding to the first network device;

means for accessing the credential repository;

means for contacting the first network device; and

means for testing the validity of the first set of credentials.

14. The system of Claim 13, wherein the credential repository further comprises, for the first network device, a plurality of a corresponding credential state, a corresponding protocol identifier, a corresponding address, a total number of instances of use of at least one credential in the first set of credentials, a corresponding candidate credential frequency counter associated with at least one credential in the first set of credentials, a recency of use of at least one credential in the first set of credentials, and the administrative locality of at least one credential in the first set of credentials.

15. The system of Claim 13, further comprising:

means for assigning a credential state to at least one credential in the first set of credentials.

16. The system of Claim 13, further comprising, when at least one credential in the first set of credentials is valid:

means for adding the at least one credential to a candidate credential queue.

17. The system of Claim 13, further comprising, when at least one credential in the first set of credentials is not valid:

means for pinging the first network device to determine whether the first network device is contactable.

18. The system of Claim 13, further comprising, when at least one credential in the first set of credentials is not valid:

second means for selecting a second set of credentials from a candidate credential queue; and

second means for testing the validity of the second set of credentials.

19. The system of Claim 18, further comprising, when at least one credential in the second set of credentials is not valid:

means for prompting a user for a third set of credentials; and

when the third set of credentials is received from the user, third means for testing the validity of the third set of credentials.

20. A system for determining one or more credentials of a network device, comprising:

a credential repository comprising a first set of credentials corresponding to a first network device; and

a credential discovery agent configured to select the first network device from among a plurality of network devices, access the credential repository, contact the first network device, and test the validity of the first set of credentials.

21. The system of Claim 20, wherein the credential repository further comprises, for the first network device, a plurality of a corresponding credential state, a corresponding protocol identifier, a corresponding address, a total number of instances of use of at least one credential in the first set of credentials, a corresponding candidate credential frequency counter associated with at least one credential in the first set of credentials, a recency of use of at least one credential in the first set of credentials, and the administrative locality of at least one credential in the first set of credentials.

22. The system of Claim 20, wherein the credential discovery agent is further configured to assign a credential state to at least one credential in the first set of credentials.

23. The system of Claim 20, wherein the credential discovery agent is further configured, when at least one credential in the first set of credentials is valid, to add the at least one credential to a candidate credential queue.

24. The system of Claim 20, wherein the credential discovery agent is further configured, when at least one credential in the first set of credentials is not valid, to ping the first network device to determine whether the first network device is contactable.

25. The system of Claim 20, wherein the credential discovery agent is further configured, when at least one credential in the first set of credentials is not valid, to select a second set of credentials from a candidate credential queue and test the validity of the second set of credentials.

26. The system of Claim 25, wherein the credential discovery agent is further configured, when at least one credential in the second set of credentials is not valid, to prompt a user for a third set of credentials and, when the third set of credentials is received from the user, to test the validity of the third set of credentials.

27. A candidate credential queue for determining a credential of a network device, comprising:

a plurality of sets of credentials, each of the plurality of sets of credentials having a corresponding ranking indicative of a probability that the plurality of sets of credentials is currently in use by the network device.

28. The candidate credential queue of Claim 27, wherein the ranking is determined based on at least one of a candidate credential frequency counter, a recency of use counter, an administrative locality associated with the corresponding set of credentials, preferences of the user, a user configured priority, and an ordering of validation of at least one set of credentials on another network device.

29. A method for determining at least one credential of a network device, comprising:

selecting at least one credential;

contacting a network device;

testing the validity of the at least one credential; and

assigning a ranking to the at least one credential based on whether or not the at least one credential is valid.

30. The method of Claim 29, wherein the ranking reflects a probability that the at least one credential is in current use in the network associated with the network device.

31. A system for determining at least one credential of a network device, comprising:

a credential discovery agent configured to assign a rank to at least one credential based on whether or not the at least one credential is valid.

32. The system of Claim 31, wherein the rank reflects a probability that the at least one credential is in current use in the network associated with the network device.

33. The system of Claim 31, wherein the credential discovery agent is further configured to select at least one credential from a candidate credential queue, test the validity of the at least one credential, and assign the rank to the at least one credential based on whether or not the at least one credential is valid.

ABSTRACT

The present invention is directed to a system and method for determining one or more credentials of a network device. The system and method select a first network device from among a plurality of network devices, access a credential repository 108, contact the first network device, and test the validity of the first set of credentials. The credential repository 108 comprises a first set of credentials corresponding to the first network device. If a user provides invalid or no credentials, a candidate credential queue 112 can be used to guess a valid second set of credentials when the first set of credentials is not valid.

M:\4366\-60\PATENT APPLICATION.DWS.frm

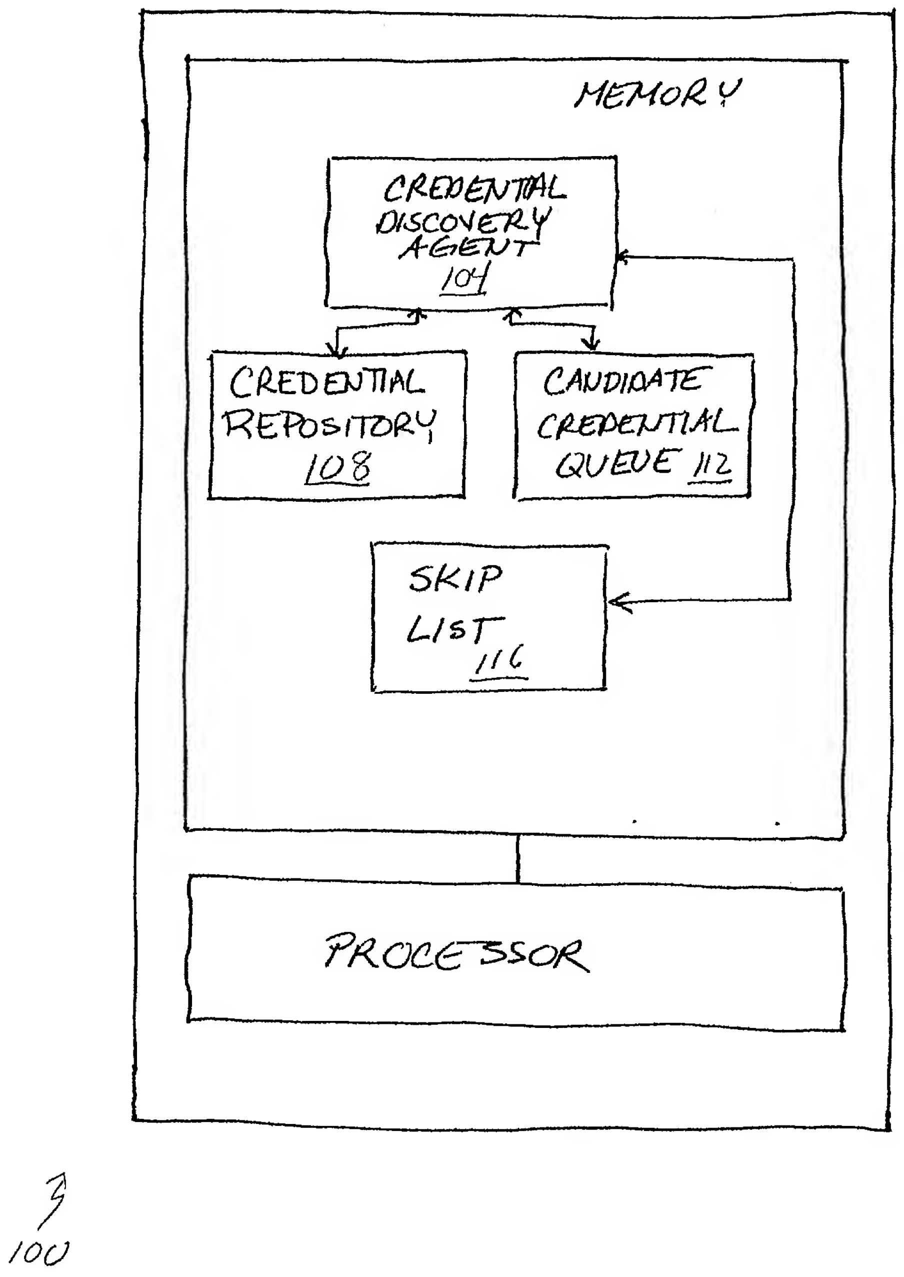

MEMORY
CREDENTIAL DISCOVERY AGENT 104
CREDENTIAL REPOSITORY 108
CANDIDATE CREDENTIAL QUEUE 112
SKIP LIST 116
PROCESSOR
100

FIG. 1

FIG. 2A

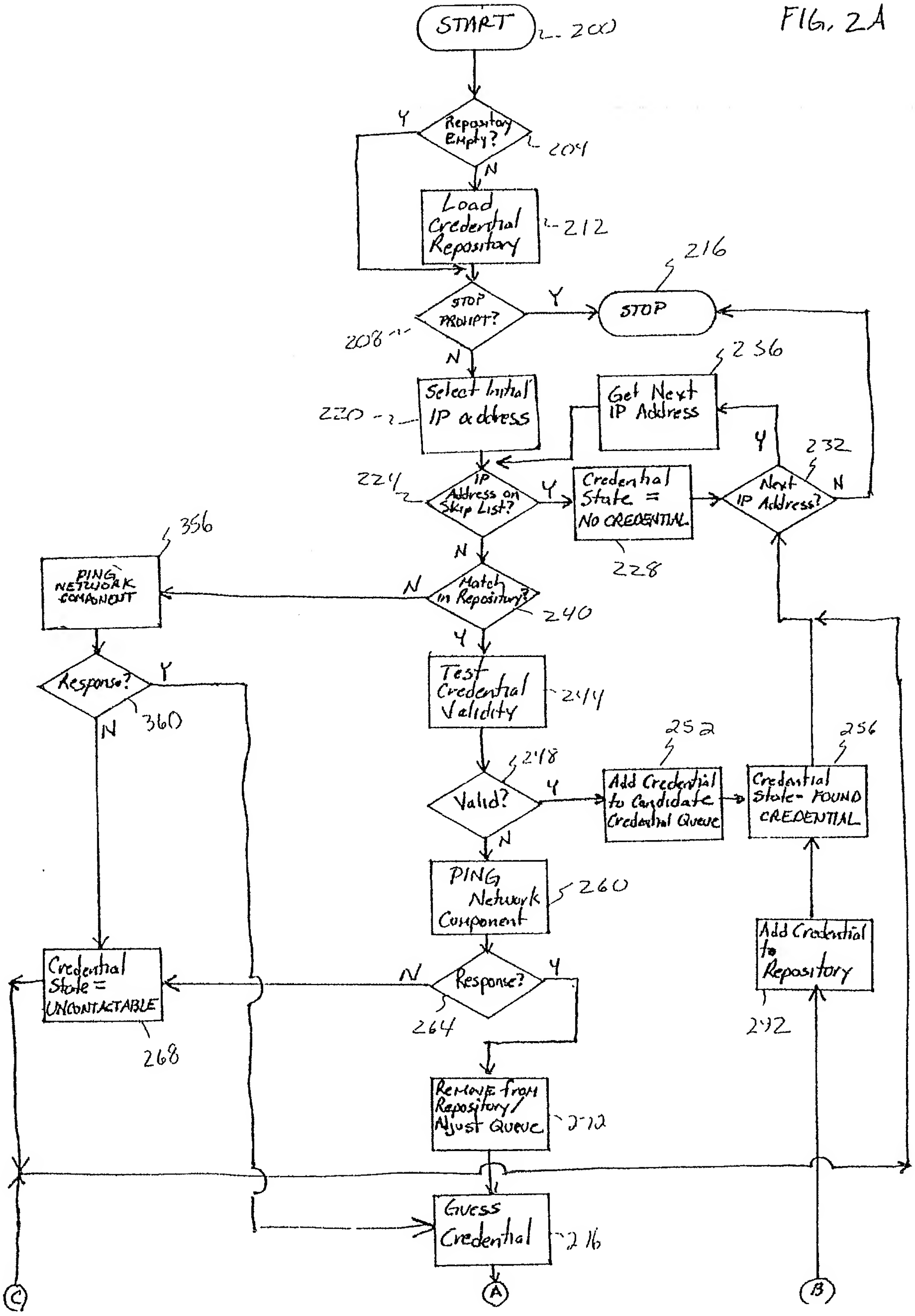
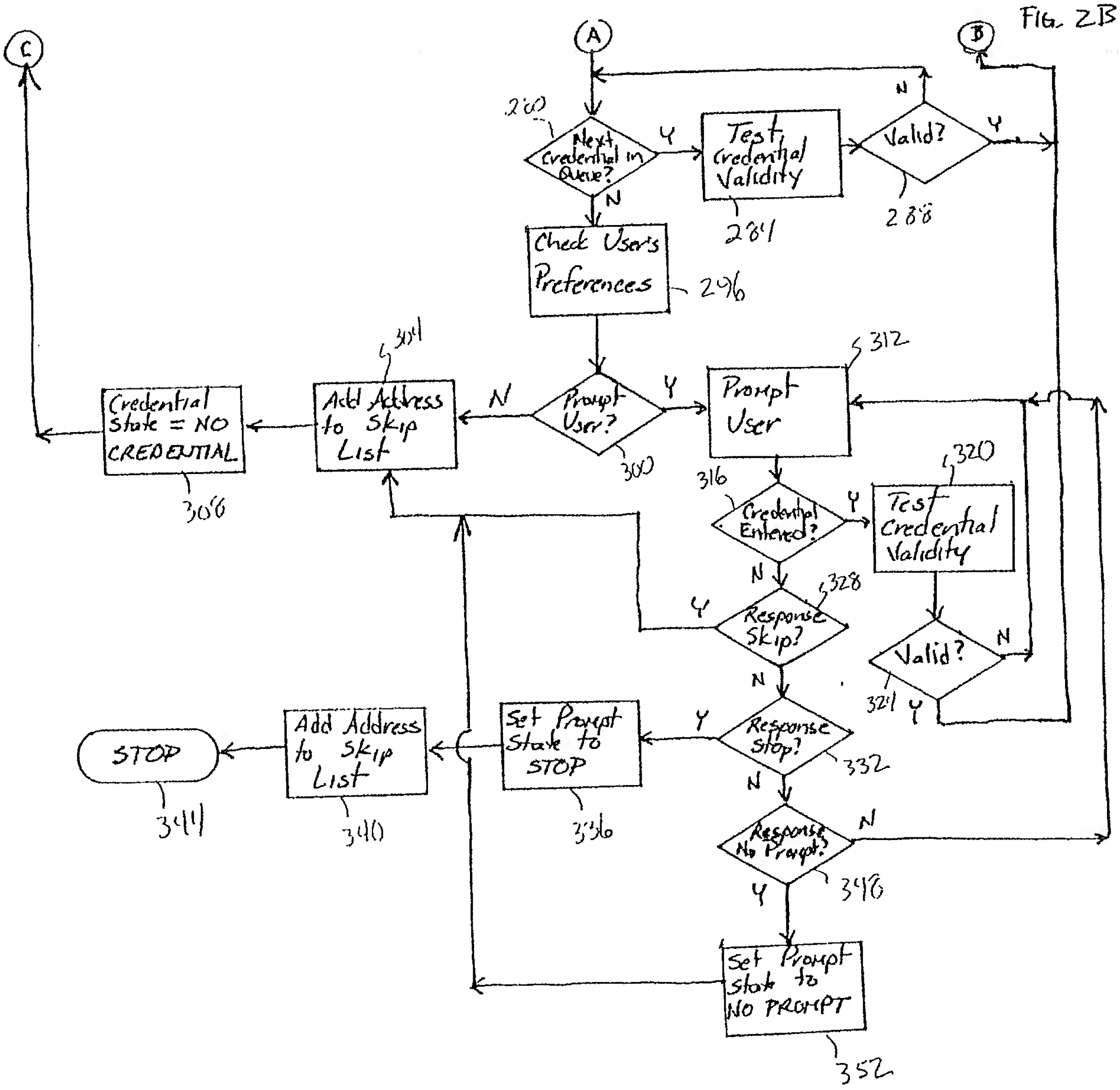

START
200
Repository Empty?
204
Y
N
Load Credential Repository
212
STOP PROMPT?
208
Y
N
STOP
216
Select Initial IP address
220
IP Address on Skip List?
224
Y
N
Credential State = NO CREDENTIAL
228
Next IP Address?
232
Y
N
Get Next IP Address
236
Match in Repository?
240
N
Y
PING NETWORK COMPONENT
356
Response?
360
Y
N
Test Credential Validity
244
Valid?
248
Y
N
Add Credential to Candidate Credential Queue
252
Credential State= FOUND CREDENTIAL
256
PING Network Component
260
Response?
264
N
Y
Credential State = UNCONTACTABLE
268
Remove from Repository/ Adjust Queue
272
Add Credential to Repository
292
Guess Credential
276
C
A
B

FIG. 2B


A
B
C
280
Next Credential in Queue?
Y
N
Test Credential Validity
284
Valid?
Y
N
288
Check User's Preferences
296
Prompt User?
Y
N
300
Add Address to Skip List
304
Credential State = NO CREDENTIAL
306
Prompt User
312
316
Credential Entered?
Y
N
Test Credential Validity
320
Valid?
N
Y
324
Response Skip?
328
Y
N
Response Stop?
332
Y
N
Set Prompt State to STOP
336
Add Address to Skip List
340
STOP
344
Response No Prompt?
348
N
Y
Set Prompt State to NO PROMPT
352